This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

❑ **BLACK BORDERS**

❑ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

❑ **FADED TEXT OR DRAWING**

❑ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

❑ **SKEWED/SLANTED IMAGES**

❑ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

❑ **GRAY SCALE DOCUMENTS**

❑ **LINES OR MARKS ON ORIGINAL DOCUMENT**

❑ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

❑ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

Rhythm®

Intelligent Planning and Scheduling Systems
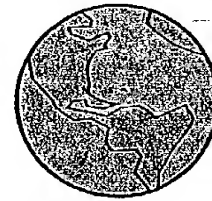
Rhythm Record Manual

i2 Technologies

Rhythm®Record Manual

i2 Technologies, Inc. (formerly Intellection, Inc.)

Release 2.8 J - Jan 31, 1996

Contents

Chapter 1

Introduction

This manual describes the set of all possible ASCII **data files** that may be input to and output from *Rhythm*® . Customers will input a subset of these files, and a subset of the fields within each file. These subsets are defined by a **spec_file**. There are different **spec_files** for each industry, and many customers further customize it to their own needs. **spec_files** are described in the *Rhythm*® *User Manual*.

Chapter 2

Rhythm Data Files

Information is input to and output from *Rhythm*® through **data files**. A **data file** consists of a series of **data records**, which each consist of a series of **data fields**. A **data field** contains a single piece of information (such as a time, a quantity, or an identifier). A **data record** collects together multiple pieces of information about a particular entity in the **factory** (such as a **resource**, a **part**, a **routing**, or an **order**). For instance, a **data record** that specifies the information about a resource may consist of **data fields** including the resource_id, the location_id, and information about how to model it such as setup rules and times, maintenance rules and times, and so on.

A **data record** is terminated by an ASCII *newline*, or *end-of-file*. A **data field** is terminated by an ASCII *tab* character, *newline*, or *end-of-file*. The set of delimiters between fields is settable with the **file.delimiters** option, which defaults to *tab*. This can be over-ridden on a file-by-file basis in the spec-file (See the *Rhythm*® *User Manual*).

Records beginning with the following characters are ignored:

- Pound-sign "**#**" (043, 35, 0x23) indicates a comment line.

- **Carriage Return** (Control-M, 015, 13, 0x0D) [i.e. empty lines]

- NewLine (Control-J, 012, 10, 0x0A)

- **NUL** (Control-@, 00, 0, 0x00)

- SUB (Control-Z, 032, 26, 0x1A)

MS-DOS terminates lines with CarriageReturn/NewLine pairs. The NewLine at the end will be ignored. DOS also terminates files with one or more Control-Z characters. These are ignored also.

2.1   Field Names

Each piece of data used, displayed, or generated by *Rhythm*® software has a particular **field name**. The **field name** is used to specify the meaning of information in a **data field** (both in **data files** and **data editors**). As such, the **field name** is used in the column titles in **data editors**. All of the **field names** defined by *Rhythm*® are listed alphabetically in the *Rhythm*® *Record Manual* with a definition of what the field means, the **field type**, related **data fields**, the **record type** in which it is defined, and the **record types** in which it is used.

2.2   Field Types

Data fields may not contain *tabs* [1] (which terminate **data fields**) or *newlines* (which terminate both **data fields** and **data records**). Any other ASCII character may appear in a **data field** [2] How those ASCII characters are interpreted by *Rhythm*© depends upon the **field type** of the **data field**, as described in the remainder of this section.

char	A single character. Trailing blanks are ignored.
Integer	A whole decimal number (positive or negative). Blank or null inputs translate to zero. Leading and trailing blanks are ignored. Trailing non-numeric characters default to being ignored, but will produce error messages if the **strict_conversion** option is turned on.
number	A real number. May be negative. Decimal point is optional. Scientific notation is accepted, using **Fortran** conventions. Blank or null inputs translate to zero. Leading and trailing blanks are ignored. Trailing non-numeric characters default to being ignored, but will produce error messages if the **strict_conversion** option is turned on.
Name	An arbitrary set of characters used to name something. Leading and trailing blanks are ignored. Names may be as long or short as you wish.
Time_Format	All **Time** and **Date** fields require a format specification. *Rhythm*© format specifications are character strings, where special formatting characters are substituted with the corresponding time elements. All other characters will be part of the resulting time string.

```
Example: specification =  "When: Ww \_D MMM YYYY, hh:mmap"
                 result =  "When: Su  1 SEP 1991, 07:23pm"
```

The special formatting characters are:

tt	Hour in 12 hour format, leading zero
_t	Hour in 12 hour format, no leading zero
hh	Hour in 24 hour format, leading zero
_h	Hour in 24 hour format, no leading zero
mm	Minute, leading zero
_m	Minute, no leading zero
ss	Second, leading zero
_s	Second, no leading zero
AP	AM/PM flag, uppercase
ap	AM/PM flag, lowercase
zzz	time zone abbreviation
DD	Day of month, leading zero
_D	Day of month, no leading zero
MM	Month of year, leading zero
_M	Month of year, no leading zero
MMM	Month Abbreviation, all caps

---

[1] The field separator defaults to *tab*, but can be changed. See the *Rhythm*© *User Manual.*

[2] If a field value begins with the pound-sign "#", and the field is the first field in the record, the record will be interpreted as a comment. To avoid this, put one or more spaces before the "#"

Mmm	Month Abbreviation. capitalized
mmm	Month Abbreviation. lowercase
MR	Rounded Month of year, leading zero (month rounded up when day not displayed and day of month is greater than 21)
_R	Rounded Month of year, no leading zero (month rounded up when day not displayed and day of month is greater than 21)
MMR	Rounded Month Abbreviation, all caps
Mmr	Rounded Month Abbreviation, capitalized
mmr	Rounded Month Abbreviation, lowercase
YY	Year mod 100
YYYY	Year
WW	2 character Day of week abbreviation, all caps
Ww	2 character Day of week abbreviation, Capitalized
ww	2 character Day of week abbreviation, lowercase
WWW	3 character Day of week abbreviation, all caps
Www	3 character Day of week abbreviation, Capitalized
www	3 character Day of week abbreviation, lowercase

Formatted_Time   A date and/or time input according to a **Time_Format**. The default format is **DDMMMYYYYhhmmss** (e.g. 25MAR1993183000 is March 25th 1993 at 6:30pm) The default may be changed via the **file.time_format** option. In addition, each **Formatted_Time** field has an associated **Time_Format** field which can be used to specify the format.

2.3 Field Classes

Many **data fields** have common meanings. This is encapsulated in the field **Class**. For example, all **data fields** which may be either true or false are in the *Boolean* class.

2.4 Req Field

The **Req** field contains one of the following keys that indicates whether or not the field is required.

Blank	Blank indicates an optional field.
P	It is a primary key, and therefore required.
U	It has no default value (uninitialized and is therefore required.
n	Positive numbers indicate secondary key fields.
-n	Negative numbers indicate fields which require the corresponding (positive) secondary field.

The fields with positive numbers are optional, but if they occur, then the fields with the same positive number are required. A field with a negative number may NOT be included unless its corresponding positively numbered field is included.

Example from Supplier_Part_Record:

Req	Comments
P	Required
P	Required
1	Optional, but required if lead_time is used
1	Optional, but required if lead_time_uom is used
2	Optional, but required if cost_uom is used
-2	Optional
	Optional

Chapter 3

Field Classes

ACM_Flag Add Cancel Modify flag is a single character field containing one of the characters "A" "C" "M" or " ". *Blank* and nothing at all default to *Add* .

Aggregate_Resource A grouping of resources for displaying aggregate workloads.

Batch_Capacity A number describing the capacity or some aspect of capacity of a batching resource.

Batch_Type An identifier for orders which can be combined into the same runtime batch. An operation of a routing can have up to four Batch_Types (see Batch_Type_Record).

Boolean A single character field. True is one of the characters "1" "T" "t" "Y" or "y". False is one of the characters "0" "F" "f" "N" or "n". The set of *TRUE* characters is settable with the *boolean.true* default. The set of *FALSE* characters is settable with the *boolean.false* default.

Boolean_Or_Blank A single character field. True is one of the characters "1" "T" "t" "Y" or "y". False is one of the characters "0" "F" "f" "N" or "n". A blank or empty value defaults to FALSE. The set of *TRUE* characters is settable with the *boolean.true* default. The set of *FALSE* characters is settable with the *boolean.false* default.

Calendar_Time Date and Time.

Customer An entity requiring the parts of zero or more Demand_Orders.

Demand_Area The Market, or source of demand for a product. Can be a factory, warehouse, customer, etc.

Demand_Order A demand of some quantity of parts for a customer by a given due date.

ECNCODE ECN Code: A single character field. One of "A" (Add), "D"(Delete), "U"(use Up), "R"(Replacement) This field may be empty (Null)

Formatted_Time The format of a date/time string. The format is usually defaulted from the file.time_format default.

Integer An integer number.

Inventory_Buffer A location used to maintain inventory. Buffers currently have no size limits.

Location	The location of one or more Resources. Transportation times established between the various locations contribute to the planned lead times between operations of each routing. In the user interface, resources are organized by location for convenience.
Manufacturing_Order	A routing (sequence of operations) planned to build some quantity of a part. The parts output by a manufacturing are fed as input to another manufacturing order or (in the case of final assembly) shipped to a customer. Manufacturing orders thus feed the requirements of demand orders. Each manufacturing order has a routing, a set of input and output parts specified in the bill of materials, and planned start times for each operation in the routing.
Material_UOM	A unit of measure used to scale Part_Quantity. The default is a one-to-one conversion.
Money	Cost. Usually accomanied by a Unit_Of_Measure.
Name	String used to identify an object.
Part_Number	The name of a raw or manufactured part.
Part_Quantity	An amount of material (number of parts).
Percentage	Float from 0.0 to 1.0
Planning_Granularity	CAO Planning granularity. Use *P* for Planner, or *S* for Scheduler.
Procurement_Inc_Lot_Size	Not yet documented.
Procurement_Max_Lot_Size	Not yet documented.
Procurement_Min_Lot_Size	Not yet documented.
Pruned_String	Character string. Leading and trailing white-space is ignored.
Resource	A machine, workcenter, tool, workcrew, fixture, or any capacity-limited resource utilized by one or more operations.
Resource_Usage	One of "machine" "aux1" "aux2" or "operators". The strings are case insensitive.
Routing	The name of a sequence of operations.
SD_Setup_Type	An type or category of operations which appears in a Setup_Matrix describing the sequence dependent setup times when changing a resource from processing one SD_Setup_Type to another.
Scalar_Time	A time duration associated with a Unit_Of_Measure field.
Setup_Matrix	A matrix of SD_Setup_Types whose values are the sequence dependent setup times when changing a resource having the matrix from processing one SD_Setup_Type to another.
Space	Storage space. Usually accomanied by a Unit_Of_Measure.
Starting_Point	CAO starting point. One of *PST_FRESH* , *EPST_FRESH* , *PST_SIMULATION* or *EPST_SIMULATION* .
String	Character string.
Sub_Resource	A Resource which is a member of an Aggregate_Resource.
Time_Format	Format for reading and writing dates and times

Time_String	Formatted date/time string. The format is usually specified by the file.time_format default.
Unit_Of_Measure	A Unit Of Measure specifies a conversion from some external unit to the internal units used by Rhythm. For example, Rhythm stores all times as seconds. The *days* unit of measure converts days to seconds, while the *hours* unit of measure converts hours to seconds. The default (blank) unit of measure multiplies all values by one.
Unsigned	A positive integer number.
Vendor_Max_Quantity	The maximum quantity a vendor can supply within a particular lead-time.
Vendor_Part	A part number or the string *ALL_RAW_MATERIALS*. *ALL_RAW_MATERIALS* refers to all raw parts in the bill of materials. Anything else is interpreted as an individual part.
char	A single character.
clockTy	Time in seconds since January 1, 1901.
number	A number, either integer or floating-point (with or without decimal point).

Chapter 4

Record Types

4.1 Aggregate_Resource_Record

This file associates Sub_Resources with an id (the Aggregate_Resource) for the purpose of displaying overall load of aggregate resources. There is one record for each Sub_Resource associated with an aggregate resource. A given Sub_Resource can only have one aggregate resource.

This file is readable.

Aggregate_Resource_Record				
Field	Req	Class	Type	Default
aggregate_resource	P	Aggregate_Resource	Name	-Required-
sub_resource	P	Sub_Resource	Name	-Required-

     aggregate_resource   A unique identifier used to represent a group of aggregate resources. The name of the aggregate resource must be defined in a file containing records of *Resource_Record* .

     sub_resource   The name of a resource defined in a file containing records of *Resource_Record* .

Chapter 4

Record Types

4.1 Aggregate_Resource_Record

This file associates Sub_Resources with an id (the Aggregate_Resource) for the purpose of displaying overall load of aggregate resources. There is one record for each Sub_Resource associated with an aggregate resource. A given Sub_Resource can only have one aggregate resource.

This file is readable.

Aggregate_Resource_Record				
Field	Req	Class	Type	Default
aggregate_resource	P	Aggregate_Resource	Name	-Required-
sub_resource	P	Sub_Resource	Name	-Required-

> **aggregate_resource**   A unique identifier used to represent a group of aggregate resources. The name of the aggregate resource must be defined in a file containing records of *Resource_Record* .

> **sub_resource**   The name of a resource defined in a file containing records of *Resource_Record* .

4.2   Anchor_Record

Each record specifies a constraining resource in the factory that should be load balanced in the automatic optimization functions.

This file is readable.

Anchor_Record				
Field	Req	Class	Type	Default
resource	P	Resource	Name	-Required-
balancing_algorithm	P	Name	Name	-Required-
ideal_utilization_level		number	number	1.0
min_utilization_level		number	number	0.0

balancing_algorithm   The method of balancing to be used at this resource is specified by balancing_type. The current methods available are:
SIMULATION: this method uses a *mini-simulation* of the processing of the operations available for this resource to generate a feasible sequence, based on the sequencing rules defined for the resource.
PULL_PUSH: this method uses the *CAO* algorithm to move operations to earlier time buckets (pull) or later time buckets (push), based on defined constraints.

ideal_utilization_level   Defines a target utilization percentage that this resource should be limited to when doing the balancing. NOTE :To specify 100% utilization user must specify 1.0 in this column, if user specifies 100 it will be equivalent to saying 1000% utilization.

min_utilization_level   Defines min target utilization percentage that resource should achieve while balancing. Its value should be between max/ideal_utilization_level and zero. (NOTE : Not available in the current version)

resource   Specifies the resource for which this data record applies.

4.3 Attributes_Record

A control file that typically contains a single record. This record specifies default attributes for the data set (plan start-time, etc.).

This file is readable.

Attributes_Record				
Field	Req	Class	Type	Default
start_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
start_time	P	Formatted_Time	Formatted_Time	-Required-

start_time Specifies the date and time that should be used as the *current time* for critical planning and scheduling calculations such as simulation and forward propagation of EPSTs.

start_time_format Format for reading and writing dates and times

4.4 Available_Capacity_Record

Records of this file define resource calendar information as entered in *Rhythm* . This file is generated and maintained only through *Rhythm* .

This file is readable.

Available_Capacity_Record				
Field	Req	Class	Type	Default
resource	U	Resource	Name	
start_time_cal_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
start_time_cal		Formatted_Time	Formatted_Time	Unknown_Time
end_time_cal_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
end_time_cal		Formatted Time	Formatted_Time	Unknown_Time
value		Percentage	number	0

end_time_cal   The format of a date/time string. The format is usually defaulted from the file.time_format default.

end_time_cal_format   Date/time format to use for end_time_cal.

resource   Name of resource having calendar information defined.

start_time_cal   The starting date for this calendar information.

start_time_cal_format   Date/time format to use for start_time_cal.

value   Percentage of total time available to resource that this interval_type takes up.

4.5 Batch_Definition_Record

Records of this file describe batches created either manually or automatically through *Rhythm* . Collections of orders (either demand or manufacturing) can be grouped together. There will be one record for each manufacturing order associated with a particular batch.

This file is readable.

Batch_Definition_Record				
Field	Req	Class	Type	Default
batch_id		Batch_Task	Name	NULL
batch_type		Batch_Type	Name	NULL
demand_order_id		Name	Name	
mfg_order_id	P	Manufacturing_Order	Name	-Required-
resource		Resource	Name	NULL
operation_id	P	Name	Name	-Required-
quantity_uom		Material_UOM	Name	Material_Quantity_UOM
quantity	P	Part_Quantity	number	-Required-
acm_flag		ACM_Flag	char	ACM_Add

acm_flag   Add Cancel Modify flag is a single character field containing one of the characters "A" "C" "M" or " ". *Blank* and nothing at all default to *Add* .

batch_id   A unique identifier for a batch. Rhythm appends the same prefix, as defined by the server default *batch_id_prefix* , to all batch ids. By default, the prefix is *B* , so batch_id should be something like *B001* .

batch_type   The type of batch created. The batch types available are:
----batch type 1---- Text for batch type 1.
----batch type 2---- Text for batch type 1.
This field is currently not used.

demand_order_id   This field is obsolete.

mfg_order_id   The manufacturing order associated with the jobs being batched.

operation_id   The operation id where this manufacturing order is performed.

quantity   The quantity of this manufacturing order. This field is currently unused.

quantity_uom   The Unit of Measure associated with quantity. This field is currently unused.

resource   The resource for which this batch information applies.

4.6   Batch_Size_Record

The batch_size record type specifies which resources are batching resources.   Each data record in this file assigns various constraints on the batch size of operations of a particular Batch_Type when run at a particular resource. *Rhythm* uses these constraints when forming orders into batches on the resource.

This file is readable.

Batch_Size_Record				
Field	Req	Class	Type	Default
type		Batch_Type	Name	0
resource	P	Resource	Name	-Required-
formula	P	Batch_Capacity_Formula	Name	-Required-
ideal_capacity	1	Part_Quantity	number	1
min_capacity		Part_Quantity	number	0
max_capacity		Part_Quantity	number	1
capacity_uom	-1	Unit_Of_Measure	Name	Material_Quantity_UOM
runtime_uom		Unit_Of_Measure	Name	BLANK_UOM
runtime		Scalar_Time	number	0
acm_flag		ACM_Flag	char	ACM_Add

acm_flag   Can be used to modify runtimes of existing batch_types at resources. Batch runtimes can vary over time so we can have a fixed batch_size file and another file to specify runtimes.

capacity_uom   The unit of measure in which *ideal_capacity* , *min_capacity* , and *max_capacity* sizes are specified. If left blank, Batch_Capacity fields are interpreted as Part_Quantity.

formula   The formula that controls use of min_capacity, max_capacity, and ideal_capacity when *Rhythm* forms operations of this type at this resource into batches. The value of this field should be *Batch_Capacity_Formula* for the default formula. Other formulas can be used in combination with C++ customizations on top of the base *Rhythm* software.

ideal_capacity   The size that the automated batching logic tries to achieve for this batch type. Additionally, when planning operations at a batching resource before specific batches have been formed, the ideal_level is used to estimate the expected load placed on the resource.

max_capacity   Specifies the maximum size constraint for batches of this type. The automated batch formation logic uses this in the generation of batches.

min_capacity   Specifies the minimum size constraint for batches of this type. The automated batch formation logic uses this in the generation of batches.

resource   The batching resource for which the record specifies sizes.

runtime   The run time that should be used for batches of this type regardless of the operation run times in the batch.

runtime_uom   The time unit of measure in which the batch *runtime* is specified.

type   The type of batch for which the record specifies sizes. This allows different batch types to have different size constraints at the same resource. If type field is missing then it means that the capacity values

are for all types at the resource.

4.7 Batch_Type_Record

Each record of the file establishes the Batch_Type(s) of one of the operations of a routing. Orders scheduled at this operation can be batched together if they have compatible Batch_Types. Batch_Types are defined in a file containing records of type Routing_Record.

This file is readable.

Batch_Type_Record				
Field	Req	Class	Type	Default
routing	P	Name	Name	-Required-
operation	P	Name	Name	-Required-
preop_time_uom	1	Unit_Of_Measure	Name	BLANK_UOM
preop_time	1	Scalar_Time	number	0
unit_runtime_uom	2	Unit_Of_Measure	Name	BLANK_UOM
unit_runtime	2	Scalar_Time	number	0
run_rate_uom	3	Material_UOM	Name	Material_Quantity_UOM
run_rate_per	3	Unit_Of_Measure	Name	BLANK_UOM
run_rate	3	Part_Quantity	number	0
base_yield		Percentage	number	1.0
cooldown_time_uom	4	Unit_Of_Measure	Name	BLANK_UOM
cooldown_time	4	Scalar_Time	number	0
part_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
transfer_batch_quantity		Part_Quantity	number	MATERIAL_QUANTITY_UNINITIALIZED
sequence_dependent_setup_type		SD_Setup_Type	Name	0
primary	5	Batch_Type	Name	0
secondary1	-5	Batch_Type	Name	0
secondary2	-5	Batch_Type	Name	0
secondary3	-5	Batch_Type	Name	0
base_machine_resource		Resource	Name	NULL
priority		Integer	Integer	unknown.INT
acm_flag		ACM_Flag	char	ACM_Add
link_downstream_p		Boolean	char	FALSE

acm_flag   Add Cancel Modify flag is a single character field containing one of the characters "A" "C" "M" or " ". *Blank* and nothing at all default to *Add*.

base_machine_resource   If this routing uses a single resource, you can specify it here. This is a convenience feature so some customers can avoid the more general *operation_resources_data* file.

base_yield   Defines the expected yield of this operation. If there is an expected spoilage of 3%, then *base_yield* would be "0.97".

cooldown_time   Specifies the time material sits after runtime. The amount of time is independent of the quantity of parts.

cooldown_time_uom   The units used for *cooldown_time* .

link_downstream_p   The link_downstream_p field indicates if this operation is linked to the following operation in this routing. If this value is *TRUE* then Rhythm will always propagate the planned times to the following operation to prevent any time gap between the two operations.

part_quantity_uom	The Unit of Measure associated with material_quantity.
operation	An operation in the operation sequence for *routing* .
primary	The primary batch type for this operation.
priority	This allows users to give priority to a particular routing and operation. CAO will use it in pull-push logic.
routing	The name of a routing which includes this *operation* . This routing id will be used by records of *Bill_Of_Materials_Records* to define where material is used when producing parts.
run_rate	Continuous flow processes use *run_rate* INSTEAD OF *unit_runtime* . *run_rate* is in terms of quantity / time. For example, 39 Tons per hour, or 19 liters per minute. The unit of measure (e.g. Tons or Liters) is specified with *run_rate_uom* field. The time unit of measure (e.g. hours or minutes) is in the *run_rate_per* field.
run_rate_per	A Unit of Measure associated with run_rate. E.g. in 19 liters per minute, this value would be "MINUTES".
run_rate_uom	A Unit of Measure associated with run_rate. E.g in 19 liters per minute, this value would be "LITERS".
secondary1	An alternate batch type for this operation.
secondary2	An alternate batch type for this operation.
secondary3	An alternate batch type for this operation.
sequence_dependent_setup_type	The type this routing puts this resource into after performing this operation. For example, a painting machine may put this resource into a state of "WHITE" or "BLACK" which is used in a setup matrix to specify sequence dependent setup time from type "WHITE" to "BLACK" and vice versa. If no type is input, then the type is set to a concatenation of operation and part. This field has meaning only if a setup matrix has been established.
transfer_batch_quantity	Transfer batch quantity for the operation.
preop_time	The amount of time materials occupy the resource before runtime. The amount of time is independent of the quantity of parts. However, if there are no input parts, this amount of time is ignored.
preop_time_uom	The units used for *preop_time* .
unit_runtime	Specifies the runtime per unit of *consumed_part_number* specified in the file containing records of *Bill_Of_Materials_Record* . For instance, if *operation* "Assemble_Table" requires "8" "legs" and "2" "table_top" and *unit_runtime* and *unit_runtime_uom* are "1" "HOURS" to produce 2 tables, and we plan an order for three tables (12 legs and 3 table_tops), the total runtime of the operation is 1.5 hours.
unit_runtime_uom	the unit of measure for *unit_runtime* .

4.8 Bill_Of_Materials_Record

Records of this file specify the bill of materials. A single bill-of-material (BOM) consists of a produced part and quantity, a set of consumed parts and quantities, and a routing. A BOM might have effectivity information as well, saying that it is valid at a given start date. Because there is one consumed part but potentially more than one input parts and routings, much of the information in this file will repeat.

Currently we assume that *consumed_part_number* is input (consumed) and *produced_part_number* is output (produced) at the first operation of the routing.

Optional fields ecn_code, ecn_date, and ecn_number establish Date and Use Effectivity for a BOM. Date effectivity constrains *Rhythm* to use the BOM only if operation PSTs can fall between two time points, the BOM start time and BOM end time. Use Effectivity defines three sets of parts, all of which are consumed parts in the bill: Use-Up Parts, Accompanying Parts, and Replacement Parts. Note that multiple level Use-Up effectivity is not supported. When *Rhythm* runs out of Use-Up Parts, it starts using Replacement Parts instead of Use-Up and Accompanying Parts. Note that our definition of Use Effectivity does not require that PSTs of operations of a use-effective BOM come before PSTs of operations of its replacement BOM. Thus, the schedule may say to start using the *new part* before the *old part* is used up.

If date effectivity fields are used, this file needs to be sorted by increasing ecn_date, blank ecn_date before nonblank values, then by ecn_code.

This file is readable.

Bill_Of_Materials_Record				
Field	**Req**	**Class**	**Type**	**Default**
produced_part_number		Part_Number	Name	NULL
new_produced_part_number		Part_Number	Name	NULL
produced_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
produced_quantity	P	Part_Quantity	number	-Required-
routing	U	Routing	Name	
operation	-1	Name	Name	
operation_sequence_number	-1	Integer	Integer	unknown.LONG
consumed_part_number		Part_Number	Name	NULL
new_consumed_part_number		Part_Number	Name	NULL
consumed_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
consumed_quantity	P	Part_Quantity	number	-Required-
acm_flag		ACM_Flag	char	ACM_Add
ecn_code	2	ECNCODE	char	ECNCODE_Null
ecn_description	-2	String		
ecn_date_format	-2	Time_Format	Time_Format	DDMMMYYYYhhmmss
ecn_date	-2	Formatted_Time	Formatted_Time	Unknown_Time
ecn_number	-2	String		
min_quantity		Part_Quantity	number	0
max_quantity		Part_Quantity	number	Infinity
multiple_quantity		Part_Quantity	number	0
quantity_constraint_type		Name	Name	*FINISHED*
flexible_min_quantity_p		Boolean	char	TRUE
mfg_consolidation_interval		number	number	0
min_fixed_shrinkage		number	number	0
min_percent_shrinkage		number	number	0
norm_fixed_shrinkage		number	number	0
norm_percent_shrinkage		number	number	0

acm_flag   Add Cancel Modify flag is a single character field containing one of the characters "A" "C" "M" or " ". *Blank* and nothing at all default to *Add* .

consumed_part_number   One of possibly several parts consumed by this bill.

consumed_quantity   The quantity of *consumed_part_number* consumed for every *produced_quantity* of *produced_part_number* . For instance, for a routing building tables, the *produced_quantity* and *produced_part_number* are "1" and "Table", while the *consumed_quantity* and *consumed_part_number* values are "4", "Legs" and "1" and "TableTop".

consumed_quantity_uom   The Unit of Measure associated with consumed_quantity.

ecn_code   If not blank, this field establishes that the consumed_part_number and consumed_quantity is qualified by date effectivity information (ecn_date) or use effectivity information (ecn_number). The *ecn_code* specifies how to apply the record:

"A" (Add): The record applies to the version of the BOM starting at *ecn_date* . If the file contains a BOM with an earlier *ecn_date* , this BOM is a copy of it with the addition of the *consumed_part_number* and *consumed_quantity* . The copying feature eliminates the need to repeat every record of the bill for every date effective change. Otherwise, this BOM is the original BOM (but is still not effective until *ecn_date*). When *ecn_date* is blank, it represents *currently effective* .

"D" (Delete): The record deletes the consumed_part_number from the BOM starting at *ecn_date* . The *consumed_part_number* is presumably a member of the previously effective BOM.

"U" (Use-Up Part): The record directs *Rhythm* to replace any Use-Up and Accompanying Parts having this record's *ecn_number* with any Replacement Parts having this record's *ecn_number* . For instance, a given BOM might specify *ecn_number* "100" with Use-Up Parts "10" and "11", Accompanying Part "12", and Replacement Parts "13", "14", and "15". It could also specify *ecn_number* "101" with Use-Up Part "20", no Accompanying Parts, and Replacement Part "21". If *Rhythm* runs out of part "10", it replaces demand for "10", "11", and "12" with parts "13", "14", and "15". Later, if it runs out of part "20", it starts using "21" instead.

"S" (Accompanying Part): The record adds *consumed_part_number* and *consumed_quantity* as an Accompanying Part for *ecn_number* . There must be at least one Use-Up Part and one Replacement Part, but Accompanying Part is optional.

"T" (Replacement Part): The record adds *consumed_part_number* and *consumed_quantity* as a Replacement Part for *ecn_number* .

ecn_date   The date and time on which this bill is effective. If blank, this bill is initially effective. Its effectivity ends only if there is another BOM with the same *produced_part_number* but with a later *ecn_date* . Thus, it is easy to provide a succession of date effectivity changes. First, place the earliest effective BOM records in the file. Then follow it with changes effective date "D1". Then follow that with changes effective date "D2", where "D1" < "D2". Each new *ecn_date* causes a copy of the BOM

records of the most recent *ecn_date* (for the given *consumed_part_number* and *routing*), so only the net changes have to be included. The earliest effective BOM becomes ineffective at "D1". The "D1" BOM becomes ineffective at "D2". If there are no other *ecn_date* entries for this *consumed_part_number* and routing, the "D2" BOM is effective to infinity.

ecn_date_format	Date/time format to use for *ecn_date* .
ecn_description	A description of this particular effectivity. This field is for UI display purposes only.
ecn_number	The tag which binds sets of Use-Up, Accompanying, and Replacement Parts (see *ecn_code*). For a given *ecn_number* , its Use-Up and Accompanying Parts are replaced by the Replacement Parts.
flexible_min_quantity_p	When the producible quantity falls below min_quantity, if this is FALSE then we set the producible quantity to 0, otherwise we leave it alone.
max_quantity	The maximum quantity of *produced_part_number* Rhythm plans for manufacturing orders of this bill. See field *quantity_constraint_type* for more information.
mfg_consolidation_interval	When positive, this value specifies an override of the -mfg_consolidation_interval command line parameter for manufacturing orders of this bill. Its units are days. See the documentation on -mfg_consolidation_interval for more information.
min_fixed_shrinkage	Indicates a minimum fixed loss of the "consumed_material_type" per mfg-order. It only applies when the part comes, at least partially, from procurements or mfg-orders whose first operation is not yet complete.
min_percent_shrinkage	Indicates a minimum percent loss of the "consumed_material_type" per mfg-order. This is similar to a maximal yield for a given "consumed_material_type" where max_yield = (1 - 0.01 * "min_percent_shrinkage") but only applies to procurements and manufactured parts whose first operation is not yet complete.
min_quantity	The minimum quantity of *produced_part_number* Rhythm plans for manufacturing orders of this bill. See field *quantity_constraint_type* for more information.
multiple_quantity	When non-zero this field constrains the quantity of *produced_part_number* Rhythm plans for manufacturing orders to a multiple of *multiple_quantity* . See field *quantity_constraint_type* for more information.
new_consumed_part_number	One of possibly several parts consumed by this bill. The *new_* prefix makes file part_number_data unnecessary to include in the data set. If part_number_data is present, your spec_file should instead use data field *consumed_part_number* .
new_produced_part_number	The part produced by this bill. This value will repeat for every *new_consumed_part_number* . The *new_* prefix makes file part_number_data unnecessary to include in the data set. If part_number_data is present, your spec_file should instead use data field *produced_part_number* .
norm_fixed_shrinkage	Indicates an expected fixed loss of the "consumed_material_type" per

	mfg-order. It only applies when the part comes, at least partially, from procurements or mfg-orders whose first operation is not yet complete.
norm_percent_shrinkage	Indicates an expected percent loss of the "consumed_material_type" per mfg-order. This is similar to a yield for a given "consumed_material_type" where the yield would be $(1 - 0.01 *$ "normal_percent_shrinkage") but only applies to procurements and manufactured parts whose first operation is not yet complete.
operation	This field is obsolete.
operation_sequence_number	This field is currently not used.
routing	The routing used to produce *produced_part_number*. This value will repeat for every *consumed_part_number*. It is possible to establish more than one routing for sets of produced and consumed parts. For instance, routing1 and routing2 might both produce Part10. The consumed parts might differ for routing1 and routing2, or they might be the same. If they are the same, the *consumed_quantity* values might differ.
produced_part_number	The part produced by this bill. This value will repeat for every *consumed_part_number*. *produced_part_number* must be defined in a file of type *Part_Number_Record*.
produced_quantity	The quantity of part produced by this bill. Commonly, this value is 1. This value will repeat for every *consumed_part_number*.
produced_quantity_uom	The Unit of Measure associated with produced_quantity.
quantity_constraint_type	Specifies whether *min_quantity*, *max_quantity*, and *multiple_quantity* fields are in terms of started (pre-routing-yield) or finished (post-routing-yield) quantities. The field is either *FINISHED* or *STARTED*. For instance, if *routing* operations have a combined yield of 50% and *min_quantity* is 100, a *quantity_constraint_type* of FINISHED specifies that the minimum produced quantity is 100 and thus the first operation must start 200. Conversely, if *quantity_constraint_type* is STARTED, the first operation must start 100 and thus the minimum produced quantity is 50.

4.9 Block_Formation_Record

An alternative to the Cycle_Calendar_Record where the data specifies production rate and size. Time intervals are calculated, assuming the records are sequential starting at start_time

This file is readable.

Block_Formation_Record				
Field	Req	Class	Type	Default
block_id	P	Integer	Integer	-Required-
cycle	P	Planning_Cycle	Name	-Required-
resource	P	Resource	Name	-Required-
block_size_uom	1	Material_UOM	Name	Material_Quantity_UOM
block_size	P	Part_Quantity	number	-Required-
rate_uom	-2	Material_UOM	Name	Material_Quantity_UOM
rate	2	Part_Quantity	number	-1
minimum_block_size_uom	-3	Material_UOM	Name	Material_Quantity_UOM
minimum_block_size	3	Part_Quantity	number	-1
start_time_format	-4	Time_Format	Time_Format	DDMMMYYYYhhmmss
start_time	4	Formatted_Time	Formatted_Time	Unknown_Time
end_time_format	-5	Time_Format	Time_Format	DDMMMYYYYhhmmss
end_time	5	Formatted_Time	Formatted_Time	Unknown_Time

block_id   Globally unique Block Identifier. A value of -1 means a unique number will be automatically generated.

block_size   The minimum amount of material allowed to be processed in this block.

block_size_uom   Units used for *block_size* .

cycle   Unique identifier Planning Cycle.

end_time   The block end date/time (write-only field)

end_time_format   Date/time format to use for end_time (write-only field)

minimum_block_size   The minimum amount of material allowed to be processed in this block. The -1 default means use the default specified in the Cycle_Resource_Record.

minimum_block_size_uom   Units used for *minimum_block_size* .

rate   The rate of production for an amount of material processed per day. The -1 default means use the default specified in the Cycle_Resource_Record.

rate_uom   Units used for *rate* .

resource   Name of resource having cycle time defined.

start_time   The block start date/time (write-only field)

start_time_format   Date/time format to use for start_time (write-only field)

4.10 CAO_Parameters_Record

A control file that typically contains a single record. This record specifies default attributes that control the behavior of the *CAO* algorithm, which focuses on constraint anchored optimization, in *Rhythm* . This file is readable.

CAO_Parameters_Record				
Field	Req	Class	Type	Default
pst_rule		Name	Name	*MIN_WIP*
convergence_speed		Integer	Integer	4
max_balancing_limit		Integer	Integer	6
starting_condition		Starting_Point	Name	PST_SIMULATION
consider_resource_constraints		Boolean	char	TRUE
no_duedate_violation		Boolean	char	FALSE
priority_model		Name	Name	*Duedate_Priority_Model*
utilization_goal_p		Boolean	char	FALSE
run_utilization_fix_pass_p		Boolean	char	TRUE
propagate_after_moving_every_task		Boolean	char	FALSE
archive_plan		Boolean	char	FALSE
diffusion		Boolean	char	TRUE
look_ahead_uom	1	Unit_Of_Measure	Name	BLANK_UOM
look_ahead	1	Scalar_Time	number	0
resource_criticality		Name	Name	*CRITICALITY_BASED*

archive_plan   A single character field. True is one of the characters "1" "T" "t" "Y" or "y". False is one of the characters "0" "F" "f" "N" or "n". The set of *TRUE* characters is settable with the *boolean.true* default. The set of *FALSE* characters is settable with the *boolean.false* default.

consider_resource_constraints   If this flag is TRUE it will consult the resource calendar and cycle calendar while propagating CEST/CLST constraints.

convergence_speed   *CAO* converges to solution. Convergence speed is a positive number indicating the how many times *CAO* will allow a task to be pulled durring balancing. In this case, pull means any number of sequential settings of PST earlier followed by setting the PST earlier.

diffusion   A single character field. True is one of the characters "1" "T" "t" "Y" or "y". False is one of the characters "0" "F" "f" "N" or "n". The set of *TRUE* characters is settable with the *boolean.true* default. The set of *FALSE* characters is settable with the *boolean.false* default.

look_ahead   The amount of time to look ahead when attempting to pull or push jobs.

look_ahead_uom   The time unit of measure in which *look_ahead* is specified.

max_balancing_limit   The maximum number of times one resource will be balanced during the run of *CAO* .

no_duedate_violation   If set to *TRUE* , *CAO* will not push the job beyond its LPST.

priority_model   This will specify the priority model used by *CAO* for pull/push criteria.

propagate_after_moving_every_task   This field is obsolete in release 2.7.

pst_rule   These rules specify how *CAO* sets the planned start time for those tasks which are moved by *CAO*. The available rules are:
JIT:
The start times of tasks moved by *CAO* will be set to finish the tasks as on time as possible if the order is early or as early as possible if the order is late.
MAX_UTIL_MIN_WIP:
This rule enforces maximum utilization while minimizing cycle time.
MIN_WIP:
CAO will try to minimize the gap of the finished time of one operation and the start time of the next operation, i.e. minimize cycle time.
OPTIMISTIC:
The start times of tasks moved by *CAO* will be set to finish the tasks as early as possible.

resource_criticality   A number representing how balancing this resource will effect the rest of the schedule. It should be related to the total number of jobs and resources that will potentially be affected during balancing.

run_utilization_fix_pass_p   A single character field. True is one of the characters "1" "T" "t" "Y" or "y". False is one of the characters "0" "F" "f" "N" or "n". The set of *TRUE* characters is settable with the *boolean.true* default. The set of *FALSE* characters is settable with the *boolean.false* default.

starting_condition   Specifies from which condition to begin CAO. It could be EPSE/PST. Available starting conditions are, PST_FRESH : Starts with PST and will also move scheduled tasks if it requires to. PST_SIMULATION : Starts with PST and will not pull/push scheduled tasks.
EPST_FRESH : Starts with EPST and will also move scheduled tasks if it requires to. EPST_SIMULATION: Start with EPST and will not pull/push scheduled tasks.

utilization_goal_p   If set to TRUE it will use minimal propagation while running *CAO*. So it will only change pst of upstream or downstream tasks if it needs to.

4.11   Cycle_Calendar_Record

Records of this file represent information entered and maintain via the *Rhythm* cycle calendar interface. Each record indicates a cycle is in effect for the specified time interval on the specified resource. It is possible to have multiple cycles in effect at the same time on a given resource. (See Cycle_Resource_Record)

This file is readable.

Cycle_Calendar_Record				
Field	Req	Class	Type	Default
cycle	P	Planning_Cycle	Name	-Required-
resource	P	Resource	Name	-Required-
start_time_cal_format	-1	Time_Format	Time_Format	DDMMMYYYYhhmmss
start_time_cal	1	Formatted_Time	Formatted_Time	Unknown_Time
end_time_cal_format	-1	Time_Format	Time_Format	DDMMMYYYYhhmmss
end_time_cal	1	Formatted_Time	Formatted_Time	Unknown_Time
start_time	2	clockTy	Unsigned	unknown.SECONDS
end_time	2	clockTy	Unsigned	unknown.SECONDS
interval_type		Name	Name	
description		String		NULL

cycle	Unique identifier for making reference to a cycle.
description	A textual description of the cycle.
end_time	Obsolete way to input start_time, kept for compatibility with old data. Will be removed in a future version (after 2_2).
end_time_cal	The end date for this cycle interval
end_time_cal_format	Date/time format to use for end_time_cal.
interval_type	Unused.
resource	Name of resource having cycle time defined.
start_time	Obsolete way to input start_time, kept for compatibility with old data. Will be removed in a future version (after 2_2).
start_time_cal	The starting date for this cycle interval
start_time_cal_format	Date/time format to use for start_time_cal.

4.12 Cycle_Resource_Record

This data file specifies the cycles that a particular resource provides. Entries in this file are required in order to use the *Rhythm Cycle Calendar* interface to specify the time intervals for the cycles. Use the Cycle_Routing_Record file to restrict which routing operations may be performed during the resource's cycle. Several fields in this record apply to a specific cycle regardless of the resource: *setup_time* , *default_rate* , and *default_minimum_block_size* . Thus last record for a specific cycle will override any earlier values for these fields from a previous record referencing the same cycle.

This file is readable.

Cycle_Resource_Record				
Field	Req	Class	Type	Default
cycle	P	Planning_Cycle	Name	-Required-
resource	P	Resource	Name	-Required-
setup_time_uom	-1	Unit_Of_Measure	Name	BLANK_UOM
setup_time	1	Scalar_Time	number	0
default_rate_uom	-2	Material_UOM	Name	Material_Quantity_UOM
default_rate	2	Part_Quantity	number	-1
default_minimum_block_size_uom	-3	Material_UOM	Name	Material_Quantity_UOM
default_minimum_block_size	3	Part_Quantity	number	-1
use_block_planning		Boolean	char	FALSE
acm_flag		ACM_Flag	char	ACM_Add

acm_flag Add Cancel Modify flag is a single character field containing one of the characters "A" "C" "M" or " ". *Blank* and nothing at all default to *Add* .

cycle Unique identifier to reference a particular cycle.

default_minimum_block_size The minimum amount of material allowed to be processed. (Only used by the Block Formation Editor.) A value of -1 indicates the default minimum will be the actual block size specified by the Block_Formation_Record or interactively using the Block Formation Editor.

default_minimum_block_size_uom Units used for *default_minimum_block_size* .

default_rate The amount of material processed per day. (Only used by the Block Formation Editor.) The -1 default means use the global default value as specified in the server default: /-bfe_default_rate/

default_rate_uom Units used for *default_rate* .

resource The resource for which this cycle information applies.

setup_time The amount of time required for setup of this resource for this cycle. If a task is requiring this cycle, the set up time is subtracted from the resource's capacity in the load graph bucket where the task begins. This setup time is only subtracted once per resource load graph bucket regardless of the number of planned tasks requiring this cycle.

setup_time_uom Units used for *setup_time* .

use_block_planning Causes tasks on this resource to be planned with the start time at the beginning of the containing block. The downstream task is constrained after the end of the containing block. The resource must have

/Block_Formation_Record/ entries. If any record sets this to TRUE, that resource will use block planning regardless of other values in other records.

4.13 Cycle_Routing_Record

A record in this data file defines which cycle to used for a particular routing operation. Tasks for operations in this file will only be planned on resources providing the same cycle as specified in the *Cycle_Resource_Record* Furthermore, the task must be planned during the cycle's time interval specified in the *Cycle_Calendar_Record* .

This file is readable.

Cycle_Routing_Record				
Field	Req	Class	Type	Default
routing	P	Routing	Name	-Required-
operation	P	Name	Name	-Required-
cycle	P	Planning_Cycle	Name	-Required-

cycle : Name of the cycle.

operation : Particular operation within a routing for which this cycle information applies.

routing : Name of the routing which includes this operation.

4.14   Demand_Order_Record

These records describe the demand orders to be filled by the factory. Each data record in this data file defines a request (demand order) for a quantity of some product along with a due date, priority, etc.

This file is readable.

Field	Req	Class	Type	Default
demand_order_id	P	Name	Name	-Required-
sales_due_date_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
sales_due_date	P	Formatted_Time	Formatted_Time	-Required-
part_number		Part_Number	Name .	NULL
new_part_number		Part_Number	Name	NULL
part_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
part_quantity	P	Part_Quantity	number	-Required-
priority		number	number	unknown.FLOAT
category		Pruned_String	Pruned_String	
customer		Customer	Name	NULL
demand_area		Demand_Area	Name	NULL
replan_order_p		Boolean	char	FALSE
ship_early		Boolean	char	FALSE
crsd_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
crsd		Formatted_Time	Formatted_Time	Unknown_Time
promise_date_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
promise_date		Formatted_Time	Formatted_Time	Unknown_Time
ship_partial		Boolean	char	FALSE
make_to_stock_p		Boolean	char	UNKNOWN_BOOLEAN
stock_buffer		Inventory_Buffer	Name	unspecified_inventory_buffer_id
acm_flag		ACM_Flag	char	ACM_Add

Table caption: **Demand_Order_Record**

acm_flag   Add Cancel Modify flag is a single character field containing one of the characters "A" "C" "M" or " ". *Blank* and nothing at all default to *Add* .

category   The type of this order. This is used to create logical groupings of the orders.

crsd   Customer Requested Ship Date: Orders with the ship_early flag False are displayed in the ship_early problem_window pane if they complete too much before CRSD, "To much" is determined by the pane's tolerance.

crsd_format   The date/time format to use in parsing *crsd* .

customer   This field identifies the requestor (customer) that placed this order.

demand_area   The plant (production location) that should be assigned to produce this order.

make_to_stock_p   A flag specifying whether this order is to be made regardless of inventory. A value of "T" means always build this part, while "F" means take it from stock if possible, build it otherwise. The default value is the value of the demand order part ""s stock_part_p flag, which

itself defaults "F".

part_quantity — The quantity of the end product (*part_number* or *new_part_number*) that this order requires.

part_quantity_uom — The unit of measure in which *part_quantity* is specified.

part_number — The end product to be produced for this order. Use *part_number* in the spec_file when warnings should be generated upon encountering a previously undefined part number. (i.e. not listed in the file containing records of Part_Number_Record.) This type should not be used when *new_part_number* is used.

new_part_number — The end product to be produced for this order. Use *new_part_number* in the spec_file when warnings should not be generated upon encountering a previously undefined part number. Use this instead of the *part_number* field when there is no part master provided. (i.e there is no file containing records of Part_Number_Record.) This type should not be used when *part_number* is used.

demand_order_id — A unique identifier which will identify this particular demand order.

priority — The priority or importance associated with this order. The priority is used by the automatic planning algorithms in *Rhythm* to compare the importance of one order relative to other orders. The higher the number, the greater the priority.

promise_date — Date promised to customer.

promise_date_format — The date/time format to use in parsing *promise_date* .

replan_order_p — A flag specifying whether or not the saved plan for this order should be ignored, thus forcing a new default plan to be generated. A value of "T" means the order should be forced to replan, while "F" means the saved plan should be restored and used, if possible.

sales_due_date — The date by which the customer was promised that the product could be produced. This is the date *Rhythm* tries to meet in its automatic planning algorithms.

sales_due_date_format — The date/time format to use in parsing *sales_due_date* .

ship_early — A flag specifying whether this order is a problem if it is ready too early. A value of "T" means earliness is a problem, while "F" (the default) means it isn't. Too early means more than some tolerance from the *crsd* , see that field.

ship_partial — A flag specifying whether this order is a problem if it is ready too early. A value of "T" means partial shipments are allowed, while "F" (the default) means they aren't.

stock_buffer — The inventory buffer destination for a part built for a demand order with make_to_stock = "T". If make_to_stock_p is "F" this field is ignored.

4.15 Dispatch_List_Record

This record supports writing of dispatch list information. User can specify format through spec file and use this record to write schedules at a particular resource or at all resources. Currently the record supports some of the important attributes of scheduled jobs but it should finally support all the fields supported in the schedule list of manual scheduler.

This file is readable.

Dispatch_List_Record				
Field	Req	Class	Type	Default
resource	U	Resource	Name	
next_resource		Resource	Name	NULL
part_number	U	Part_Number	Name	
operation_id		Name	Name	NULL
quantity	U	Part_Quantity	number	
runtime_uom		Unit_Of_Measure	Name	BLANK_UOM
runtime		Scalar_Time	number	0
formatted_runtime		Pruned_String	Pruned_String	
pst_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
pst		Formatted_Time	Formatted_Time	Unknown_Time
schedule_end_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
schedule_end_time		Formatted_Time	Formatted_Time	Unknown_Time
lpst_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
lpst		Formatted_Time	Formatted_Time	Unknown_Time
customer	U	Customer	Name	
due_date_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
due_date		Formatted_Time	Formatted_Time	Unknown_Time
order_id		Name	Name	NULL
batch_id		Name	Name	NULL

batch_id It the batch id of the batch, which this scheduled job belongs to.

customer This field is ignored on input. When command line option mfg_consolidation_horizon is zero, each operation has only one demand order, and the demand orders customer is written in this field. Otherwise each operation has potentially many demand orders, so a blank is written.

due_date This field is ignored on input. When command line option mfg_consolidation_horizon is zero, each operation has only one demand order, and the demand orders due date is written in this field. Otherwise each operation has potentially many demand orders, and the earliest due date among the demand orders is written.

due_date_format Time format for the demand order due date.

formatted_runtime Runtime of the scheduled operation broken down into components: Days:Hours:Minutes:Seconds.

lpst It the latest possible start time that this job should have started to meet the order due date. It is the operation due date.

lpst_format LPST time format.

next_resource	This information indicates, where is this job to go after this operation is completed.
operation_id	The operation within a routing.
order_id	It is the manufacturing order id to which this scheduled job belongs to.
part_number	Part produced by this manufacturing order.
pst	Start time of the scheduled operation.
pst_format	Time format for printing start time of the scheduled operation.
quantity	It is the quantity produced by this scheduled operation.
resource	It is the work center at which this scheduled operation will be performed.
runtime	Runtime of the scheduled operation.
runtime_uom	The time unit of measure in which the batch *runtime* is specified.
schedule_end_time	End time of the scheduled operation.
schedule_end_time_format	Time format for printing end time of the scheduled operation.

4.16 Dispatch_Rule_Record

Records of this file are used to effect how *CAO* operates. The dispatching method for each resource and its accompanying arrival time is defined. The following dispatching algorithms are available:

FIFO:

Select the first job (from the Ready queue).

SRPF:

Select the job with shortest remaining process time. The remaining process time is the estimated critical path cycle time.

MIN_SETUP:

Select the job with the same setup type as the previous job. If none exists, then select the first job.

FIX_SIZE:

Select the first job and, when the resource batch capacity allows, other jobs with the same batch type.

ANCHOR_FIFO:

Same as FIFO, except that when the Ready queue is empty, select the first job whose EPST is less than the simulator current time from the Waiting queue.

ANCHOR_SRPF:

Select the job with the shortest remaining process time only if it is needed by other resources (i.e. constraint times not empty).

ANCHOR_MIN_SETUP:

Select a job with the same setup type as the previous job. If such a job can not be found in the Ready queue, search in the Waiting queue. The selected job from the Waiting queue should have EPST less than the simulator current time and PST no further than lookahead_length from the simulator current time. If no job can be found, the first job from the Ready queue is selected.

The following are available for arrival time:

BUCKET_BASED:

CEST:

Jobs will arrive at their CESTs.

PST:

Jobs will arrive at their PSTs.

This file is readable.

Dispatch_Rule_Record				
Field	**Req**	**Class**	**Type**	**Default**
resource	U	Resource	Name	
dispatching_algorithm		Name	Name	NULL
arrival_time		Name	Name	NULL

arrival_time   One of the above defined arrival times.

dispatching_algorithm   One of the above defined dispatching algorithms.

resource   The resource having its dispatching algorithm defined.

4.17 Dynamic_CAO_Parameters_Record

This file is read in before every run of *CAO* . It contains the parameters of *CAO* that can be changed between the runs of *CAO* . The file should specify the parameters that can change (or that the user wants to change) between two runs of *CAO* . The file is read before every run of *CAO* , while the *CAO_Parameters* file is only read once at initialization. Ideally, a user should be able to change all the dynamic parameters through the UI, but many times an option is not added to the UI until later. If the UI allowed changes of all the parameters in the Dynamic_CAO_Parameters file, then that file would not be added.

This file is readable.

Dynamic_CAO_Parameters_Record				
Field	Req	Class	Type	Default
priority_model		Name	Name	*Duedate_Priority_Model*
run_utilization_fix_pass_p	P	Boolean	char	-Required-
propagate_after_moving_every_task	P	Boolean	char	-Required-
archive_plan	P	Boolean	char	-Required-
diffusion	P	Boolean	char	-Required-
resource_criticality	P	Name	Name	-Required-
niterations		Integer	Integer	2.0

archive_plan If TRUE, *Rhythm* will archive the plan before the run of *CAO* . User will be able to *Undo CAO* if archive_plan flag is set to TRUE. More over when user runs *CAO* multiple times, *Rhythm* will always first undo the *CAO* go back to the archived plan. In cases where this flag is set to FALSE *CAO* will start on the current plan and if user wants to *Undo CAO* s/he needs to generate plan again. For larger data sets it is advised to set this flag to FALSE.

diffusion If set to TRUE, *CAO* will use diffusion based balancer else it will use look ahead based balancer. For detail information on *diffusion*, or look ahead based balancer, refer to the *Rhythm User Manual* .

niterations If *CAO* is using user defined search path then this field will specify the number of iterations *CAO* will go over that search path.

priority_model This will specify the priority model used by *CAO* for pull/push criteria.

propagate_after_moving_every_task This field is obsolete in release 2.7.

resource_criticality A number representing how balancing this resource will effect the rest of the schedule. It should be related to the total number of jobs and resources that will potentially be affected during balancing.

run_utilization_fix_pass_p If TRUE, *CAO* will run second pass at the end to fix remaining capacity shortage problems. While running second pass *CAO* will relax the constraints and allow balancer to create WIP. User can set this flag through command line and by default its TRUE.

4.18 Fixed_Capacity_Bucket_Size_Record

This record allows the definition of a number of fixed-length bucket sizes used for rough scheduling purposes and many graphical display elements (e.g. load graphs). The overall scheduling horizon cannot be affected by this record.

This record is a sub-set of what can be defined by records of *Variable_Capacity_Bucket_Size_Record* .

This file is readable.

Fixed_Capacity_Bucket_Size_Record				
Field	Req	Class	Type	Default
bucket_size_uom	P	Unit_Of_Measure	Name	-Required-
bucket_size	P	Scalar_Time	number	-Required-

bucket_size   The size of the bucket. For example given a *bucket_size* of "1", and a *bucket_size_uom* of "WEEKS", the size of each bucket will be one week.

bucket_size_uom   The time units used for *bucket_size* .

4.19   Interplant_Demand_Keys_Record

This file is readable.

Interplant_Demand_Keys_Record				
Field	**Req**	**Class**	**Type**	**Default**
demanding_order	P	Name	Name	-Required-
operation		Name	Name	NULL
consumer	P	Name	Name	-Required-
part	P	Name	Name	-Required-

consumer   The plant in the Interplant network demanding *part* from the supplier. The possible names are established by *Supplier_Record* s. *Consumer* is treated as the customer field of the generated Interplant demand order.

demanding_order   The demand order or manufacturing order of *consumer* needing *part* .

operation   If Blank this field indicates that *demanding_order* is a demand order of *consumer* . Otherwise *demanding_order* is a manufacturing order of *consumer* , and this field is the first operation of the manufacturing order. Note that *Rhythm* does not yet allow reservations for operations other than the first one in the routing.

part   The part required by *demanding_order* from the supplier. The supplier's name for the part is used in this field if it differs from this *Rhythm* server's name for it. Differences in part names between *Rhythm* servers in an Interplant network are expressed using the *vendor_part* field of *Vendor_Record* .

4.20   Interplant_Order_Record

This file is maintained by *Rhythm* for use in saving and restoring demand orders which fill Interplant demands. Interplant plants supplying parts to other plants save the demand orders generated.

This file is readable.

Interplant_Order_Record				
Field	**Req**	**Class**	**Type**	**Default**
demanding_order	P	Name	Name	-Required-
operation		Name	Name	NULL
consumer	P	Name	Name	-Required-
part	P	Name	Name	-Required-
supplying_order	P	Demand_Order	Name	-Required-

consumer The plant in the Interplant network demanding *part* from the supplier. The possible names are established by *Supplier_Record* s. *Consumer* is treated as the customer field of the generated Interplant demand order.

demanding_order The demand order or manufacturing order of *consumer* needing *part* .

operation If Blank this field indicates that *demanding_order* is a demand order of *consumer* . Otherwise *demanding_order* is a manufacturing order of *consumer* , and this field is the first operation of the manufacturing order. Note that *Rhythm* does not yet allow reservations for operations other than the first one in the routing.

part The part required by *demanding_order* from the supplier. The supplier's name for the part is used in this field if it differs from this *Rhythm* server's name for it. Differences in part names between *Rhythm* servers in an Interplant network are expressed using the *vendor_part* field of *Vendor_Record* .

supplying_order the demand order of this Interplant supplier plant which was generated in response to an Interplant demand from some consumer plant (through an *Interplant_Procurement_Record*). Depending on option settings, the supplying_order id is either a generated id the same as demanding_order.

4.21 Interplant_Procurement_Record

Files of this type are maintained by *Rhythm* for use in saving and reading Interplant demands and responses. *interplant_data_PLANTNAME* is a template, not a data file. The actual data files are named *interplant_data_* concatenated with a plant name established by *Supplier_Record*.

Using this record, Interplant demands are issued by one *Rhythm* server (the consumer) to communicate demand for a part to another *Rhythm* server (the supplier). An Interplant response is issued by the supplier back to the consumer, communicating how many parts can be supplied and at what time they are estimated to be available.

This file is readable.

Interplant_Procurement_Record				
Field	Req	Class	Type	Default
demanding_order	P	Name	Name	-Required-
operation		Name	Name	NULL
consumer	P	Name	Name	-Required-
part	P	Name	Name	-Required-
is_demand_p	P	Boolean	char	-Required-
time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
time	P	Formatted_Time	Formatted_Time	-Required-
quantity_uom		Material_UOM	Name	Material_Quantity_UOM
quantity	P	Part_Quantity	number	-Required-
priority		number	number	unknown.FLOAT
supplier	P	Name	Name	-Required-

consumer The plant in the Interplant network demanding *part* from the supplier. The possible names are established by *Supplier_Record* s. *Consumer* is treated as the customer field of the generated Interplant demand order.

demanding_order The demand order or manufacturing order of *consumer* needing *part*.

is_demand_p If TRUE, this record describes an Interplant demand order for *part* from *supplier* requested by *consumer*. If FALSE, this record describes the response to such a demand by *supplier* to *consumer*. The response tells when the parts can be supplied and how many can be supplied.

operation If Blank this field indicates that *demanding_order* is a demand order of *consumer*. Otherwise *demanding_order* is a manufacturing order of *consumer*, and this field is the first operation of the manufacturing order. Note that *Rhythm* does not yet allow reservations for operations other than the first one in the routing.

part The part required by *demanding_order* from the supplier. The supplier's name for the part is used in this field if it differs from this *Rhythm* server's name for it. Differences in part names between *Rhythm* servers in an Interplant network are expressed using the *vendor_part* field of *Vendor_Record*.

priority If *is_demand_p* is TRUE, this number is the priority *consumer* requests *supplier* to set for the demand order which will supply *part*. Otherwise, this field is ignored.

quantity The quantity of *part* demanded when is_demand_p is TRUE, else the quantity of *part* supplied by *supplier* via *supplying_order*.

quantity_uom	The Unit of Measure associated with the *quantity* field.
supplier	The plant in the Interplant network supplying *part* to *consumer*. The possible names are established by *Supplier_Record* s.
time	If is_demand_p is TRUE, this is the time that *demanding_order* needs *part*. Otherwise it is the time parts are available from *supplying_order*.
time_format	Date/time format to use for the *time* field.

4.22 Inventory_Buffer_Record

Each record of the file specifies the *location* for an *inventory_buffer*. The location is used to account for the transportation time between the inventory buffers and other locations in the factory model.

This is an optional file.

This file is readable.

Inventory_Buffer_Record				
Field	Req	Class	Type	Default
inventory_buffer	P	Inventory_Buffer	Name	-Required-
location		Location	Name	Location::unspecified( )

inventory_buffer A unique identifier naming the particular inventory buffer whose location is being defined.

location The location of this inventory buffer.

4.23   Lot_Reassignment_Record

Used to keep track of separate quantities of material processed at an operation, such as coils in the metals industry. This file is written out by Rhythm, primarily in order to keep track of the results of splitting orders by lots.

A lot reassignment record tells where the lot is (manufacturing order only). This information is written out by Rhythm. The point is to keep track of the location of lots which, because of a split, are no longer where the lot record says they are.

When Rhythm starts up, the lot data is read first and the lots are placed accordingly. Then the lot reassignment data is read. At this time, if a lot has been assigned to order A but belongs to order B according to the lot reassignment data, then the lot will be moved from A to B, but only if order B is the result of a split from lot A, or A and B were both split from the same order.

This file is readable.

Lot_Reassignment_Record				
Field	**Req**	**Class**	**Type**	**Default**
manufacturing_order	P	Name	Name	-Required-
lot_id		Name	Name	NULL

> **lot_id**   Identifier for this lot
>
> **manufacturing_order**   The unique manufacturing order id used to associate this lot with a particular demand order.

4.24   Lot_Record

Used to keep track of separate quantities of material processed at an operation, such as coils in the metals industry. A lot record tells where the lot is (manufacturing order and operation) and what is the quantity of material in the lot. This information is supplied by the user. If lot_on_remaining_ops is TRUE, then the user will also see the lot at all operations downstream from the one where it really is.

This file is readable.

Lot_Record				
Field	**Req**	**Class**	**Type**	**Default**
manufacturing_order		Name	Name	NULL
current_operation		Name	Name	NULL
lot_on_remaining_ops		Boolean	char	FALSE
lot_id	P	Name	Name	-Required-
lot_qty	U	Part_Quantity	number	

current_operation	The operation for this lot within a particular routing.
lot_id	Identifier for this lot
lot_on_remaining_ops	If true, lot is also placed an all tasks downstream of this one, but this task is the actual location of the lot.
lot_qty	Quantity of material in this lot.
manufacturing_order	The unique manufacturing order id used to associate this lot with a particular demand order.

operation   An operation of *routing* (if present). Helps specify which material is loaded on *resource* (see Cases #1 and #4 above).

operation_start_time   NOTE: this field is currently not supported. The time at which material loaded on *resource* started being processed. If blank we assume that it is being started at *current time* .

produced_part   Helps specify which material is loaded on *resource* (see Case #4 above).

resource   The resource whose state is described by this record.

routing   Helps specify which material is loaded on *resource* (see Case #4 above).

start_time_format   Date/time format to use for operation_start_time.

4.26 Make_To_Stock_Record

Records of this file describe the most recent action of *Repopulate Stock Buffers*. There will be one record for each make_to_stock demand order that gets more than zero of its part made.

This file is readable.

<table>
<tr><th colspan="5">Make_To_Stock_Record</th></tr>
<tr><th>Field</th><th>Req</th><th>Class</th><th>Type</th><th>Default</th></tr>
<tr><td>producing_order</td><td>P</td><td>Name</td><td>Name</td><td>-Required-</td></tr>
<tr><td>inventory_buffer</td><td></td><td>Inventory_Buffer</td><td>Name</td><td>unspecified_inventory_buffer_id</td></tr>
<tr><td>part_number</td><td>P</td><td>Part_Number</td><td>Name</td><td>-Required-</td></tr>
<tr><td>quantity</td><td>P</td><td>number</td><td>number</td><td>-Required-</td></tr>
<tr><td>arrival_time_format</td><td></td><td>Time_Format</td><td>Time_Format</td><td>DDMMMYYYYhhmmss</td></tr>
<tr><td>arrival_time</td><td>P</td><td>Formatted_Time</td><td>Formatted_Time</td><td>-Required-</td></tr>
</table>

arrival_time The time at which this part will be available in *inventory_buffer*.

arrival_time_format The date/time format to use in parsing *arrival_time*.

inventory_buffer The holding queue in which this part is sitting. If left blank it defaults to the buffer specified by command line option -unspecified_inventory_buffer_id.

part_number The part number for the quantity of parts that are being specified.

producing_order An identifier specifying with the make to stock demand order.

quantity The amount of the part produced.

4.27 Manufacturing_Order_Pegging_Record

This file associates demand orders with their final assembly manufacturing orders, and manufacturing orders with their subassembly manufacturing orders. A demand order can be fed parts by zero or one manufacturing order, and a manufacturing order can be fed parts by zero or more manufacturing orders. (Other part requirements are met from unassigned inventory or vendors.)

This file is typically written only by *Rhythm* . It is read by *Rhythm* to restore previous plans.

This file is readable.

Manufacturing_Order_Pegging_Record				
Field	**Req**	**Class**	**Type**	**Default**
produced_order	P	Name	Name	-Required-
demand_order_p		Boolean	char	FALSE
consumed_order	P	Name	Name	-Required-
quantity_fed		number	number	0
acm_flag		ACM_Flag	char	ACM_Add

acm_flag   Add Cancel Modify flag is a single character field containing one of the characters "A" "C" "M" or " ". *Blank* and nothing at all default to *Add* .

consumed_order   A manufacturing order which feeds parts to produced_order.

demand_order_p   A flag specifying whether *produced_order* is a demand order or manufacturing order.

produced_order   An identifier specifying either a demand order or manufacturing order.

quantity_fed   Quantity of material transfered from sub_project to project.

4.28 Manufacturing_Order_Record

This file specifies all manufacturing orders and associated data. Each final assembly and sub-assembly is a manufacturing order.

The association between demand order and its final assembly manufacturing order, and between a manufacturing order and its associated sub-assembly manufacturing orders is maintained in the file containing records of *Manufacturing_Order_Pegging_Record* .

This file is readable.

Manufacturing_Order_Record				
Field	Req	Class	Type	Default
manufacturing_order	P	Name	Name	-Required-
routing	P	Routing	Name	-Required-
specific_routing_id	P	Part_Number	Name	-Required-
order_quantity_satisfied		number	number	0
split_quantity		number	number	0
mfg_plant_id		Name	Name	0
acm_flag		ACM_Flag	char	ACM_Add
ecn_date_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
ecn_date		Formatted_Time	Formatted_Time	Unknown_Time

acm_flag	Add Cancel Modify flag is a single character field containing one of the characters "A" "C" "M" or " ". *Blank* and nothing at all default to *Add* .
ecn_date	Routing date effectivity.
ecn_date_format	Date/time format to use for ecn_date
manufacturing_order	The unique id used to represent a particular manufacturing order.
mfg_plant_id	Name of plant where this MFG order will be/is being produced.
order_quantity_satisfied	Quantity of the demand order this order satisfies.
routing	The routing where this order will be performed.
specific_routing_id	Determines specific Routing. Usually a bill of material is identified by a routing and the BOM produced part. Therefore, this field is usually the BOM produced part number.
split_quantity	Quantity of an order that is to be split off to make a new order.

4.29 Mfg_Order_Output_Record

This file specifies the format in which the manufacturing orders planned by *Rhythm* are output. See *Manufacturing_Order_Record* for more information about manufacturing orders.

This file is readable.

Mfg_Order_Output_Record				
Field	Req	Class	Type	Default
mfg_order		Name	Name	NULL
routing	U	Routing	Name	
primary_output_part	U	Part_Number	Name	
output_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
output_quantity	U	Part_Quantity	number	
whole_output_quantity		Integer	Integer	0
status		Pruned_String	Pruned_String	
pst_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
pst		Formatted_Time	Formatted_Time	Unknown_Time
pet_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
pet		Formatted_Time	Formatted_Time	Unknown_Time
demand_order	U	Demand_Order	Name	
within_mfg_consolidation_horizon		Boolean	char	FALSE

demand_order   A demand of some quantity of parts for a customer by a given due date.

mfg_order   A unique id used to identify a particular manufacturing order.

output_quantity   The quantity of parts in units of *output_quantity_uom* produced by this manufacturing order.

output_quantity_uom   The Unit of Measure associated with output_quantity

pet   The Planned End Time of this manufacturing order.

pet_format   Date/time format to use for pet.

primary_output_part   The part number being produced by this manufacturing order.

pst   The Planned Start Time for this manufacturing order.

pst_format   Date/time format to use for pst.

routing   The routing used to by this manufacturing order.

status   The status of this manufacturing order.

whole_output_quantity   output quantity rounded to a whole number

within_mfg_consolidation_horizon   Flag indicating whether this manufacturing order falls within the time specified by options -mfg_consolidation_horizon and -current_time.

4.30   Monthly_Production_By_Part_Record

This file is written by *Rhythm* for use by the customer in writing production by part. Production based upon forecasts for the following months needs to be output from *Rhythm* for use in financial comparison reports to be generated. After a plan has been generated, the production for each PSF and / or table in each month is output to this file.

Every month, projected production needs to be analyzed to understand the anticipated plant performance and comparison to historical performance. Since this analysis is financial, the standard costs and accounting factors are applied based on part number, and hence, production numbers that are provided are broken into monthly part numbers (consolidated over the different lots run during the month).

The information that is acquired is the total quantity of product planned over the entire month, for each month of the plan. This requires a two step process. First, each planned completion date has only month and year information. After sorting by year, month, and part number, the orders with similar part ids falling in the same month are consolidated. By saving the plan, the file will be automatically generated.

This file is readable.

Monthly_Production_By_Part_Record				
Field	Req	Class	Type	Default
ordered_part	U	Part_Number	Name	
order_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
order_quantity	U	Part_Quantity	number	
time_produced_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
time_produced		Formatted_Time	Formatted_Time	Unknown_Time
start_of_bucket_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
start_of_bucket		Formatted_Time	Formatted_Time	Unknown_Time

order_quantity	An amount of material (number of parts). Always with a Unit_Of_Measure field.
order_quantity_uom	The Unit of Measure associated with order_quantity.
ordered_part	The part number being ordered for this demand order.
start_of_bucket	The start of bucket for the procurement.
start_of_bucket_format	Date/time format to use for needed_time
time_produced	The format of a date/time string. The format is usually defaulted from the file.time_format default.
time_produced_format	Date/time format to use for when part is produced.

4.31 Operation_Operators_Record

This file is obsolete and should no longer be used as of *Rhythm 2.2*

This file is readable.

Operation_Operators_Record				
Field	Req	Class	Type	Default
routing	P	Routing	Name	-Required-
operation	P	Name	Name	-Required-
operator_group	P	Resource	Name	-Required-
min_num_operators		Integer	Integer	1
max_num_operators		Integer	Integer	1
operator_dependent_p		Boolean	char	TRUE
usage_percent		Percentage	number	1

max_num_operators

min_num_operators

operation

operator_dependent_p

operator_group

routing

usage_percent

4.32 Operation_Resources_Record

This file is optional and is only appropriate for operations which require a combination of different types of resources (such as a machine, tool, and workcrew) to run. Some factory operations may require several resource types but have so much capacity at one of these types that, for convenience, it can be ignored and left out of the set. For instance, if the factory operations require machines and workcrews but workcrew capacity is abundant, this file is not required at all, since the file containing records of *Routing_Record* is sufficient for establishing the machines (primary resources) which can run each operation.

It is entirely possible that only a few operations of the factory require records in this file, since they are the only operations which require multiple resources.

Each record of the file specifies one alternative resource for one of the required simultaneous resources. Each record has a type field which specifies which of the simultaneous resource requirements it models. It may be one of the following: "machine", "aux1", "aux2", or "operators". The four *factor* fields specify how those characteristics of the operation are affected by choosing one resource versus another. For instance, if one "machine" resource has an *op_time_factor* of "1", and another "machine" resource has an *op_time_factor* of "2", then performing the operation on the latter resource will take twice as long as on the former.

This file is readable.

Operation_Resources_Record				
Field	**Req**	**Class**	**Type**	**Default**
routing	P	Routing	Name	-Required-
operation	P	Name	Name	-Required-
simultaneous_usage	1	Resource_Usage	Resource_Usage	USAGE_MACHINE
resource	U	Resource	Name	
primary_p		Boolean	char	FALSE
preop_time_factor		number	number	1
runtime_factor		number	number	1
cooldown_time_factor		number	number	1
yield_factor		number	number	1
usage_percent		number	number	1
num_operators	-1	Integer	Integer	0

cooldown_time_factor Factor for cooldown time.

num_operators Meaningful only if simultaneous_usage is equal to "operators". The number of operators needed at this resource group.

operation The operation within the routing where this assignment applies.

primary_p True if resource defined is the primary (first one picked by *Rhythm*).

routing The routing for which a simultaneous resource, alternate resource, or an operator resource is being defined.

resource The name of the resource for simultaneous or alternate usage.

runtime_factor Factor for run time.

simultaneous_usage The simultaneous group for which definition applies. Can be any of:
"machine" The primary group.
"aux1" A secondary resource requirement.
"aux2" An additional resource requirement.
"operators" An operator resource requirement.

preop_time_factor Factor for setup time.

usage_percent   How much of this resource's time is used relative to the primary resource.

yield_factor   Factor for yield.

4.33   Part_Number_Record

This file specifies various types of parts and materials present in the factory, including products produced and sold, purchased parts consumed in producing products, and consumable resources which drive production.

Each record specifies the attributes of a particular *part_number*. *sellable?* is 1 or 0. If it is 1, *part_number* is considered a product sold externally. Usually a sellable part is the output part of at least one record of bill_of_materials_data. Unit_space and unit_space_uom are used to calculate batch capacity of a resource relative to this part.

This file is readable.

Part_Number_Record				
Field	Req	Class	Type	Default
part_number	P	Part_Number	Name	-Required-
demand_part_number		Name	Name	NULL
product_family		Name	Name	
part_number_type		Name	Name	NULL
customer_name		Name	Name	NULL
sellable_p		Boolean	char	TRUE
unit_price	1	Money	number	0
unit_price_uom	-1	Unit_Of_Measure	Name	BLANK_UOM
unit_cost	6	Money	number	0
unit_cost_uom	'6	Unit_Of_Measure	Name	BLANK_UOM
unit_space_uom	-2	Unit_Of_Measure	Name	BLANK_UOM
unit_space	2	Space	number	0
quantity_per_unit_space	-2	Part_Quantity	number	0.0
description		Name	Name	
min_routing_time_uom	3	Unit_Of_Measure	Name	BLANK_UOM
min_routing_time	3	Scalar_Time	number	0
stock_buffer		Inventory_Buffer	Name	unspecified_inventory_buffer_id
stock_part_p		Boolean	char	FALSE
acm_flag		ACM_Flag	char	ACM_Add

acm_flag Add Cancel Modify flag is a single character field containing one of the characters "A" "C" "M" or " ". *Blank* and nothing at all default to *Add* .

customer_name In certain cases the name of the customer who might have caused this part to be in the dataset will be present. The reason for this field is that sometimes there are raw materials in inventory which are not used by any BOM. The excess inventory windows display the excess raw materials that are on hand for a particular customer. To associate a customer with these raw materials which have no connection to an order (which is where customer usually lives), the customer name field is needed for the part object. If not present this defaults to an empty string.

demand_part_number A catalog identifier used by demand orders to refer to this part. This identifier is the one used in the demand order to refer to this part. If left blank it defaults to *part_number* .

description A description for the part.

part_number	The unique identifier for the part that is being specified.
min_routing_time	An estimate of the minimum time required to create any quantity of this part, given the required input parts. This is a generic lead time for the part. If the calculated production time is less than this number, this number is used instead.
min_routing_time_uom	The time unit of measure in which *min_routing_time* is specified.
part_number_type	Defines a category for the part. (e.g. Raw Material, Manufactured, etc.).
product_family	The product family this part belongs to.
quantity_per_unit_space	The quantity of this part that can be stored in one unit of space. This field is currently not used.
sellable_p	Indicates whether this part can be sold. It is possible to be both sellable and purchasable at the same time. When it is true, this part is considered a product, but it can still be used as input to a routing to produce other parts.
stock_buffer	The inventory buffer destination for a stock_part. This provides a default value for make_to_stock demand orders"" stock_buffer.
stock_part_p	A flag specifying whether this part should always be taken from inventory for use as a component in parts which contain it, or could be built if there is insufficient inventory. A value of "T" means never build this part as a component, while "F" means take it from stock if possible, build it otherwise. The default value is "F".
unit_cost	Cost. Usually accomanied by a Unit_Of_Measure.
unit_cost_uom	A Unit Of Measure specifies a conversion from some external unit to the internal units used by Rhythm. For example, Rhythm stores all times as seconds. The *days* unit of measure converts days to seconds, while the *hours* unit of measure converts hours to seconds. The default (blank) unit of measure multiplies all values by one.
unit_price	The price for one unit (quantity of 1) of this part. This field is meaningless and we err on public references whenever *sellable_p* is false. This field is currently not used.
unit_price_uom	The money unit of measure in which *unit_price* is specified.
unit_space	The amount of space occupied by one unit (quantity of 1) of this part.
unit_space_uom	The space unit of measure in which *unit_space* is specified.

4.34 Plan_Expedite_Task_Record

This file is maintained by *Rhythm* for use in saving and restoring plans. Records of this file specify manufacturing orders which have been expedited.

This file is readable.

Plan_Expedite_Task_Record				
Field	Req	Class	Type	Default
mfg_order_id		Name	Name	NULL
operation_id		Name	Name	NULL
cycle_override_p		Boolean	char	FALSE
move_time_override		number	number	0.0
queue_time_override		number	number	0.0
setup_time_override		number	number	0.0
run_time_override		number	number	0.0
wait_time_override		number	number	0.0
downstream_mfg_order		Manufacturing_Order	Name	NULL

cycle_override_p	If this is TRUE, RHYTHM will ignore the the cycle/block constraints during planning.
downstream_mfg_order	This field is blank if the operation is not the last operation of the routing. Otherwise this field gives the downstream manufacturing order for the given move_time_override. When manufacturing orders are consolidated, this information is critical since the operation will have a seperate expeditable transportation time for each fed manufacturing order.
mfg_order_id	The name of the manufacturing order which has been expedited.
move_time_override	override value for move time for expedition.
operation_id	The operation of the routing for which this manufacturing order applies.
queue_time_override	override value for queue time for expedition.
run_time_override	override value for run time for expedition. Not Implemented Yet.
setup_time_override	override value for setup time for expedition. Not Implemented Yet.
wait_time_override	override value for wait time for expedition.

4.35 Plan_Inventory_Assignments_Record

This file is maintained by *Rhythm* for use in saving and restoring plans. Records of this file specify how inventory has been assigned to demand orders and manufacturing orders.

This file is readable.

Plan_Inventory_Assignments Record				
Field	Req	Class	Type	Default
demand_or_mfg_order_id	P	Name	Name	-Required-
operation_id		Name	Name	NULL
assigned_part	P	Part_Number	Name	-Required-
assigned_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
assigned_quantity	P	Part_Quantity	number	-Required-
part_source		Name	Name	NULL
vendor_p		Boolean	char	FALSE
time_available_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
time_available	P	Formatted_Time	Formatted_Time	-Required-
time_needed_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
time_needed		Formatted_Time	Formatted_Time	Unknown_Time
time_previously_reserved		Formatted_Time	Formatted_Time	Unknown_Time
purchase_order_id		Name	Name	{unspecified}
part_number_type		Name	Name	

assigned_part	The part assigned to this order.
assigned_quantity	The quantity of parts assigned.
assigned_quantity_uom	The Unit of Measure associated with assigned_quantity.
demand_or_mfg_order_id	The demand or manufacturing order for this inventory assignment.
operation_id	The operation id of the order if the order id is a manufacturing order. If the operation_id is not present, then it is assumed that the material is for the first operation. If the assigned_part is not the same as the consumed material in the first operation, the record is ignored.
part_number_type	The type of assigned_part. This data is not used on input.
part_source	This is either an inventory buffer or vendor id. Specifies from where the material came.
purchase_order_id	If this is unassigned inventory, the purchase order associated with the inventory. If this is a procurement the field is meaningless.
time_available	The time inventory for this order is available from the part_source.
time_available_format	Date/time format to use for time_available.
time_needed	The time inventory for this order is needed. This time is the LPST of the operation or, for end item inventory assignments, the demand order due date. This data is not used on input, since LPST and due date can change.
time_needed_format	Date/time format to use for time_needed.
time_previously_reserved	The time this part was reserved.
vendor_p	Flag specifying whether order came from a vendor or not. "T" means *part_source* is a vendor,

"F" means inventory buffer.

4.36   Plan_Record

This file is maintained by *Rhythm* for use in saving and restoring plans. It stores operation PSTs and other data. The default name for the actual file is planned_start_times. See the std_spec_file for a list of the fields here that are likely to actually be included in the data.

This file is readable.

Plan_Record				
Field	**Req**	**Class**	**Type**	**Default**
mfg_order_id		Name	Name	NULL
primary_resource		Resource	Name	NULL
operation_id	P	Name	Name	-Required-
operators	1	Resource	Name	NULL
priority		Integer	Integer	unknown.INT
num_operators	-1	Integer	Integer	0
planned_start_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
planned_start_time		Formatted_Time	Formatted_Time	Unknown_Time
planned_end_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
planned_end_time		Formatted_Time	Formatted_Time	Unknown_Time
planned_transit_ready_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
planned_transit_ready_time		Formatted_Time	Formatted_Time	Unknown_Time
epst_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
epst		Formatted_Time	Formatted_Time	Unknown_Time
lpst_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
lpst		Formatted_Time	Formatted_Time	Unknown_Time
clst_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
clst		Formatted_Time	Formatted_Time	Unknown_Time
cest_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
cest		Formatted_Time	Formatted_Time	Unknown_Time
lock_count		Integer	Integer	0
planned_input_qty_uom		Material_UOM	Name	Material_Quantity_UOM
planned_input_qty		Part_Quantity	number	0
planned_output_qty_uom		Material_UOM	Name	Material_Quantity_UOM
planned_output_qty		Part_Quantity	number	0
planned_runtime_uom		Unit_Of_Measure	Name	BLANK_UOM
planned_runtime		Scalar_Time	number	0
stretched_runtime		Scalar_Time	number	0
primary_output_part		Part_Number	Name	NULL

cest   Constrained Earliest Start Time for this operation. This is calculated by forward propagation from the PST of the upstream operation. This field is output only.

cest_format   Date/time format to use for CEST.

clst   Constrained Latest Start Time for this operation. This is calculated by backward propagation from the PST of the down stream operation. This field is output only.

clst_format   Date/time format to use for CLST.

epst   Earliest Possible Start Time of this operation. This is constrained by

parts availability, original plan starting date and completion of upstream operations.

epst_format   Date/time format to use for EPST.

lock_count   An integer number.

lpst   Latest Possible Start Time for this operation. This is constrained by backward propagation from the due date for a demand order.

lpst_format   Date/time format to use for LPST.

mfg_order_id   A unique identifier referring to a manufacturing order.

num_operators   The number of individual pooled operators assigned to this operation.

operation_id   An operation within the routing of this manufacturing order.

operators   The operators resource used for this operation.

planned_end_time   The planned (or scheduled) ending time for this operation.

planned_end_time_format   Date/time format to use for planned_end_time.

planned_input_qty   The planned quantity of input parts associated with the operation.

planned_input_qty_uom   The Unit of Measure associated with planned_input_qty.

planned_output_qty   The planned quantity of output parts to be generated by this operation.

planned_output_qty_uom   The Unit of Measure associated with planned_output_qty.

planned_runtime   The planned runtime of this operation.

planned_runtime_uom   The units of measure used for *planned_runtime* .

planned_start_time   The planned start time for this operation.

planned_start_time_format   Date/time format to use for planned_start_time.

planned_transit_ready_time   The time when the output material from this operation will be ready for transit. This is the planned (or scheduled) ending time for the operation plus the cooldown time for the operation. This value is ignored during input.

planned_transit_ready_time_format   The date and time format to use for planned_transit_ready_time.

primary_output_part   The part number being produced by this manufacturing order.

primary_resource   The primary resource on which operation_id is performed.

priority   This saves out the priority of the operation in cases the user has modified it and would like to retrieve it.

stretched_runtime   The planned runtime of this operation stretched based on resource availability.

4.37   Plan_Resources_Record

This file is maintained by *Rhythm* for use in saving and restoring plans. Records of this file specify the planned assignment of resources for manufacturing orders. The particular resource used could be any of a number of possible alternate resources or the primary resource.

This file is readable.

Plan_Resources_Record				
Field	**Req**	**Class**	**Type**	**Default**
mfg_order_id		Name	Name	NULL
operation_id		Name	Name	NULL
resource	U	Resource	Name	
index		Integer	Integer	0
granularity	U	Planning_Granularity	char	

granularity   The granularity at which this manufacturing order has been planned.
P means Planner_Granularity,
S means Scheduler_Granularity.

index   An index into the set of simultaneous resource requirements.
"0" means machine resource,
"1" means aux1,
"2" means aux2, etc.

mfg_order_id   The manufacturing order id having its planned resource defined.

operation_id   The operation within a routing for which this manufacturing order applies.

resource   machine or aux resource but not operators resource. Operators are stored in Plan_Record.

4.38 Priority_Model_Record

This file will allow users to specify their own customized priority models.

This file is readable.

Priority_Model_Record				
Field	Req	Class	Type	Default
resource	U	Resource	Name	
priority_component		Name	Name	NULL
coefficient	U	number	number	

coefficient   This will be the multiplier to the priority_component in the priority function. User can specify positive (+ve) for addition or (-ve) for subtraction.

priority_component   Id of the priority component. Following components are currently available, LPST : PST : ORDER_PRIORITY : OPR_PRIORITY :

resource   The resource having its priority model defined.

4.39 Procurement_By_Part_Record

This file is written by *Rhythm* for use by the customer in reading planned procurements.

This file is readable.

Procurement_By_Part_Record				
Field	Req	Class	Type	Default
ordered_part	U	Part_Number	Name	
order_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
order_quantity	U	Part_Quantity	number	
vendor	U	Vendor	Name	
available_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
available_time		Formatted_Time	Formatted_Time	Unknown_Time
needed_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
needed_time		Formatted_Time	Formatted_Time	Unknown_Time
start_of_bucket_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
start_of_bucket		Formatted_Time	Formatted_Time	Unknown_Time
po_release_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
po_release_time		Formatted_Time	Formatted_Time	Unknown_Time

available_time	The available time for this part.
available_time_format	Date/time format to use for available_time.
needed_time	The needed time for this part. This is the planned start time of the manufacturing order"s first task"s planned start time.
needed_time_format	Date/time format to use for needed_time.
order_quantity	The quantity of parts in units of *output_quantity_uom* ordered for this demand order.
order_quantity_uom	The Unit of Measure associated with order_quantity.
ordered_part	The demand or manufacturing order directly fed by this procurement. The part number being ordered for this demand order.
po_release_time	The purchase order release time for this part. It is the needed_time minus the vendor lead time.
po_release_time_format	Date/time format to use for po_release_time.
start_of_bucket	The start of bucket for the procurement.
start_of_bucket_format	Date/time format to use for needed_time
vendor	The Vendor Name who supplies this part.

4.40 Procurement_Output_Record

This file is readable.

Procurement_Output_Record				
Field	Req	Class	Type	Default
demand_order_id	U	Demand_Order	Name	
part_number	U	Part_Number	Name	
quantity_uom		Material_UOM	Name	Material_Quantity_UOM
quantity	U	Part_Quantity	number	
supplier		Name	Name	NULL
time_available_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
time_available		Formatted_Time	Formatted_Time	Unknown_Time
time_needed_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
time_needed		Formatted_Time	Formatted_Time	Unknown_Time

part_number	The part number being procured.
demand_order_id	The name of the order for this procurement.
quantity	The quantity of parts being procured.
quantity_uom	The Unit of Measure associated with quantity.
supplier	The name of the vendor for *part_number* .
time_available	The date and time when the parts will be available.
time_available_format	Date/time format to use for *time_available* .
time_needed	The time this order needs this material to complete on time.
time_needed_format	Date/time format to use for *time_needed* .

4.41   Procurement_Record

This file is written by *Rhythm* for use by the customer in reading planned procurements.

This file is readable.

Procurement Record				
Field	Req	Class	Type	Default
demand_order_id		Name	Name	NULL
demand_or_mfg_order		Name	Name	NULL
ordered_part	U	Part_Number	Name	
order_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
order_quantity	U	Part_Quantity	number	
vendor	U	Vendor	Name	
available_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
available_time		Formatted_Time	Formatted_Time	Unknown_Time
needed_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
needed_time		Formatted_Time	Formatted_Time	Unknown_Time
latest_needed_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
latest_needed_time		Formatted_Time	Formatted_Time	Unknown_Time
po_release_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
po_release_time		Formatted_Time	Formatted_Time	Unknown_Time

available_time	The available time for this part.
available_time_format	Date/time format to use for available_time.
demand_or_mfg_order	The demand or manufacturing order directly fed by this procurement.
demand_order_id	The demand order ultimately fed by this procurement, if mfg order consolidation is not specified. Otherwise this field is blank because potentially many demand orders are ultimately fed by this procurement.
latest_needed_time	The latest needed time for this part. This is the LPST of the first operation of the the manufacturing order producing this part.
latest_needed_time_format	Date/time format to use for latest_needed_time.
needed_time	The needed time for this part.
needed_time_format	Date/time format to use for needed_time.
order_quantity	The quantity of parts in units of *output_quantity_uom* ordered for this demand order.
order_quantity_uom	The Unit of Measure associated with order_quantity.
ordered_part	The part number being ordered for this demand order.
po_release_time	The purchase order release time for this part. It is the needed_time minus the vendor lead time.
po_release_time_format	Date/time format to use for po_release_time.
vendor	The Vendor Name who supplies this part.

4.42 Production_Schedule_Record

This file is maintained by rhythm to save and restore detail schedules (sequences) at different resources in the factory. Record of this file specifies information that *Rhythm* saves for every job scheduled using detail scheduler within factory.

This file is readable.

Production_Schedule_Record				
Field	Req	Class	Type	Default
resource	U	Resource	Name	
mfg_order_id		Name	Name	NULL
operation_id		Name	Name	NULL
batch_id		Name	Name	NULL
primary_output_part	U	Part_Number	Name	
planned_input_qty_uom	-1	Material_UOM	Name	Material_Quantity_UOM
planned_input_qty	1	Part_Quantity	number	0
planned_output_qty_uom		Material_UOM	Name	Material_Quantity_UOM
planned_output_qty	U	Part_Quantity	number	
planned_start_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
planned_start_time		Formatted_Time	Formatted_Time	Unknown_Time
planned_end_time_format	-2	Time_Format	Time_Format	DDMMMYYYYhhmmss
planned_end_time	2	Formatted_Time	Formatted_Time	Unknown_Time
frozen_flag		Boolean	char	FALSE
scheduler_comments		Name	Name	

batch_id	Batch id if the scheduled manufacturing order belongs to batch and the batch is scheduled. Batch will have record for each manufacturing order that belongs to batch.
frozen_flag	If True, it means that the PST of the task is locked and is part of the frozen schedule at the given resource. Not yet supported.
mfg_order_id	Manufacturing order id of the scheduled job.
operation_id	Operation id of the scheduled job.
planned_end_time	The scheduled end time for this operation (write-only field)
planned_end_time_format	Date/time format to use for planned_end_time (write-only field)
planned_input_qty	Scheduled planned input quantity.
planned_input_qty_uom	Scheduled planned input quantity uom.
planned_output_qty	Scheduled planned output quantity of the output part.
planned_output_qty_uom	Scheduled planned output quantity uom.
planned_start_time	The scheduled start time for this operation.
planned_start_time_format	Date/time format to use for planned_start_time.
primary_output_part	Primary output part produced by this manufacturing order.
resource	Primary resource at which job is scheduled.
scheduler_comments	Special comments from the scheduler on the scheduled job.

4.43 Random_Orders_Record

This record causes *Rhythm* to generate random orders in the system. This can be used to test the system or determine the effect of load on the plant given random orders.

This file is readable.

Random_Orders_Record				
Field	Req	Class	Type	Default
demand_order_id	P	Demand_Order	Name	-Required-
number_of_orders_to_generate	P	Integer	Integer	-Required-
min_due_date_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
min_due_date		Formatted_Time	Formatted_Time	Unknown_Time
max_due_date_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
max_due_date		Formatted_Time	Formatted_Time	Unknown_Time
min_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
min_quantity	U	Part_Quantity	number	
max_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
max_quantity	U	Part_Quantity	number	

max_due_date	The latest due date for a randomly generated order.
max_due_date_format	Date/time format to use for max_due_date.
max_quantity	The maximum quantity of a randomly generated order.
max_quantity_uom	The Unit of Measure associated with max_quantity.
min_due_date	The earliest due date for a randomly generated order.
min_due_date_format	Date/time format to use for min_due_date.
min_quantity	The minimum quantity value of a randomly generated order.
min_quantity_uom	The Unit of Measure associated with min_quantity.
number_of_orders_to_generate	The number of random orders to generate.
demand_order_id	An identifier used for the randomly generated order.

4.44   Random_Unassigned_Inventory_Record

Records of this file cause *Rhythm* to generate random unassigned inventory. This can be used to test the system or determine the effect of load given random inventory.

This file is readable.

Random_Unassigned_Inventory_Record				
Field	Req	Class	Type	Default
inventory_buffer	P	Inventory_Buffer	Name	-Required-
part_number	P	Part_Number	Name	-Required-
number_of_records_to_generate	P	Integer	Integer	-Required-
min_material_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
min_part_quantity	U	Part_Quantity	number	
max_material_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
max_part_quantity	U	Part_Quantity	number	
min_arrival_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
min_arrival_time		Formatted_Time	Formatted_Time	Unknown_Time
max_arrival_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
max_arrival_time		Formatted_Time	Formatted_Time	Unknown_Time

inventory_buffer	The location where the random inventory will be placed.
part_number	The part number for which to generate random unassigned inventory.
max_arrival_time	The latest date to generate randomly. This time represents arrival at "inventory_buffer".
max_arrival_time_format	Date/time format to use for max_arrival_time.
max_part_quantity	The maximum quantity of *part_number* to be put in unassigned inventory.
max_material_quantity_uom	The Unit of Measure associated with max_material_quantity.
min_arrival_time	The earliest date to generate randomly. This time represents arrival at "inventory_buffer".
min_arrival_time_format	Date/time format to use for min_arrival_time.
min_part_quantity	The minimum quantity of *part_number* to be put in unassigned inventory.
min_material_quantity_uom	The Unit of Measure associated with min_material_quantity.
number_of_records_to_generate	The number of records to generate.

4.45 Resource_Calendar_Record

Records of this file define resource calendar information as entered in *Rhythm* . This file is generated and maintained only through *Rhythm* . Each record specifies an interval determined by the start_time_cal and end_time_cal fields, and specifies for that interval the type of capacity determined by the interval_type field. This file is readable.

Resource_Calendar_Record				
Field	Req	Class	Type	Default
resource	P	Resource	Name	-Required-
start_time_cal_format	-1	Time_Format	Time_Format	DDMMMYYYYhhmmss
start_time_cal	1	Formatted_Time	Formatted_Time	Unknown_Time
end_time_cal_format	-1	Time_Format	Time_Format	DDMMMYYYYhhmmss
end_time_cal	1	Formatted_Time	Formatted_Time	Unknown_Time
start_time	2	clockTy	Unsigned	unknown.SECONDS
end_time	2	clockTy	Unsigned	unknown.SECONDS
interval_type		Name	Name	
value		Percentage	number	0

end_time	Obsolete way to input start_time, kept for compatibility with old data. Will be removed in a future version (after 2.2).
end_time_cal	The end time of the interval specified by this record.
end_time_cal_format	Date/time format to use for end_time_cal.
interval_type	textual value specifying the category of calendar information. The possible values are: External_Type:Theoretical Capacity External_Type:Unavailable Capacity External_Type:Planned Maintenance External_Type:Rework Capacity
resource	Name of resource having calendar information defined.
start_time	Obsolete way to input start_time, kept for compatibility with old data. Will be removed in a future version (after 2.2).
start_time_cal	The start time of the interval specified by this record.
start_time_cal_format	Date/time format to use for start_time_cal.
value	Percentage of the capacity specified by interval_type that resource has during this interval. For instance, if interval_type is Unavailable Capacity and value is .25, then the resource has .25 unavailable capacity, which means it has .75 of its ordinary capacity within this interval. In this case an operation which would take 75 minutes if the resource was totally available would take 100 minutes if scheduled totally within this interval. For a pooled resource, this number will be interpreted as number of available subresources. For instance, if a pooled operator has ten individual operators, interval_type is Unavailable Capacity and value is .4, then there will be six operators available during the interval.

4.46 Resource_Layout_Record

This file supports the graphical layout of resources in the main window.

This file is readable.

Resource_Layout_Record				
Field	Req	Class	Type	Default
resource	P	Resource	Name	-Required-
x	P	Integer	Integer	-Required-
y	P	Integer	Integer	-Required-
width	P	Integer	Integer	-Required-
height	P	Integer	Integer	-Required-

height   The height of this resource's graphical representation.

resource   A unique identifier for this resource.

width   The width of this resource's graphical representation.

x   The horizontal position of this resource in its location. X values increase from left to right.

y   The vertical position of this resource in its location. Y values increase from top to bottom.

4.47 Resource_Lock_Record

Store bucket lock information. The stored date indicate all buckets before and including that time will be locked when the server is restarted.

This file is readable.

Field	Req	Class	Type	Default
Resource_Lock_Record				
resource	P	Resource	Name	-Required-
lock_horizon_format	-1	Time_Format	Time_Format	DDMMMYYYYhhmmss
lock_horizon	1	Formatted_Time	Formatted_Time	Unknown_Time

lock_horizon	The time in before which all buckets are locked.
lock_horizon_format	Date/time format to use for start_time_cal.
resource	Name of resource with locked buckets

batch_capacity_uom	This field is obsolete and should not be used.
batching_horizon	The amount of time before and after a PST used to determine whether other jobs with similar materials can be grouped with the job currently being considered for scheduling as a batch.
batching_horizon_uom	The units used for *batching_horizon* .
batching_lookahead	The time frame into which the automated batching should look to find tasks to pull for creating a batch. In other words, how far in advance of a PST should *Rhythm* look to find tasks to contribute to a batch. Orders will not be pulled from beyond the horizon if lookahead exceeds the horizon and a batch near the end of the horizon is not yet full.
batching_lookahead_uom	The units used for *batching_lookahead* .
default_setup_time	The time to setup from and source setup type to any destination setup type not specified in the file containing records of *Sequence_Dependent_Setup_Time_Record* .
default_setup_time_uom	The units used for *default_setup_time* .
estimated_queue_time	The estimated time parts wait in queue before being processed by this resource.
estimated_queue_time_uom	The time units used for the estimated queue time.
ideal_utilization_level	the ideal utilization that the resource must try to reach in each capacity_model bucket
last_refurbish_date	The date this resource was last refurbished (had preventive maintenance applied to it). This field is currently not used.
last_refurbish_date_format	Date/time format to use for *last_refurbish_date* .
lateness_tolerance	The amount of time for determining whether an order is late/early/on time in the *By Lateness* representation of the load graph. If the PST is greater than the LPST + tolerance, then the order is late. If the PST is less than the LPST - tolerance, then the order is early.
lateness_tolerance_uom	The units used for defining lateness tolerance
lifetime	The MTBF. This field is currently not used.
lifetime_uom	The units used for *lifetime* . This field is currently not used.
location	The location of this resource in the factory for fixed location resources.
maximum_utilization_level	the maximum utilization that the resource can reach in each capacity_model bucket
minimum_queue_time	The minimum time parts wait in queue before being processed by this resource. The value must be less than or equal to the estimated queue time. If -1 (the default), minimum_queue_time defaults to the value of estimated_queue_time.
minimum_queue_time_uom	The time units used for the minimum queue time.
minimum_utilization_level	the minimum utilization that the resource can reach in each capacity_model bucket

model_name Aggregate_Resource to create an aggregate resource. Sub_Resource to create a subresource. Other values are ignored and an ordinary resource is created unless number_of_pooled_resources is greater than 1.

name A unique identifier for this resource.

number_of_pooled_resources If this number is greater than 1, then this resource will be created as a pooled resource consisting of this number of subresources.

offloadable_from_p True if resource should be considered for automated offloading.

offloadable_to_p if True other resources can offload to this resource.

refurbish_time The amount of time since this resource was last refurbished (preventive maintenance).
This field is currently not used.

refurbish_time_uom The units used for *refurbish_time* .
This field is currently not used.

setup_matrix_id The name of the matrix used by this resource to determine sequence dependent setup information. This name is defined in the file containing records of *Sequence_Dependent_Setup_Time_Record* .

4.49   Routing_Record

Each record of the file specifies a routing. Each record defines the next *operation* in the sequence of operations for a *routing*. It also specifies routing related times for this operation (runtime, preop time, and cooldown time). These times can be modified when the operation runs on particular resources via the file containing records of *Operation_Resources_Record*.

Note: some assembly customer databases might not have routing ids if each part number is produced by only one routing. These applications should pass the part number id itself as *routing*.

This file is readable.

Routing_Record				
Field	Req	Class	Type	Default
routing	P	Name	Name	-Required-
operation	P	Name	Name	-Required-
preop_time_uom	1	Unit_Of_Measure	Name	BLANK_UOM
preop_time	1	Scalar_Time	number	0
unit_runtime_uom	2	Unit_Of_Measure	Name	BLANK_UOM
unit_runtime	2	Scalar_Time	number	0
run_rate_uom	3	Material_UOM	Name	Material_Quantity_UOM
run_rate_per	3	Unit_Of_Measure	Name	BLANK_UOM
run_rate	3	Part_Quantity	number	0
base_yield		Percentage	number	1.0
cooldown_time_uom	4	Unit_Of_Measure	Name	BLANK_UOM
cooldown_time	4	Scalar_Time	number	0
part_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
transfer_batch_quantity		Part_Quantity	number	MATERIAL_QUANTITY_UNINITIALIZED
sequence_dependent_setup_type		SD_Setup_Type	Name	0
primary	5	Batch_Type	Name	0
secondary1	-5	Batch_Type	Name	0
secondary2	-5	Batch_Type	Name	0
secondary3	-5	Batch_Type	Name	0
base_machine_resource		Resource	Name	NULL
priority		Integer	Integer	unknown.INT
acm_flag		ACM_Flag	char	ACM_Add
link_downstream_p		Boolean	char	FALSE

acm_flag   Add Cancel Modify flag is a single character field containing one of the characters "A" "C" "M" or " ". *Blank* and nothing at all default to *Add*.

base_machine_resource   If this routing uses a single resource, you can specify it here. This is a convenience feature so some customers can avoid the more general *operation_resources_data* file.

base_yield   Defines the expected yield of this operation. If there is an expected spoilage of 3%, then *base_yield* would be "0.97".

cooldown_time   Specifies the time material sits after runtime. The amount of time is independent of the quantity of parts.

cooldown_time_uom   The units used for *cooldown_time*.

link_downstream_p   The link_downstream_p field indicates if this operation is linked to the following operation in this routing. If this value is *TRUE* then Rhythm will always propagate the planned times to the following operation to prevent any time gap between the two operations.

part_quantity_uom   The Unit of Measure associated with material_quantity.

operation   An operation in the operation sequence for *routing* .

primary   The primary batch type for this operation.

priority   This allows users to give priority to a particular routing and operation. CAO will use it in pull-push logic.

routing   The name of a routing which includes this *operation* . This routing id will be used by records of *Bill_Of_Materials_Records* to define where material is used when producing parts.

run_rate   Continuous flow processes use *run_rate* INSTEAD OF *unit_runtime* . *run_rate* is in terms of quantity / time. For example, 39 Tons per hour, or 19 liters per minute. The unit of measure (e.g. Tons or Liters) is specified with *run_rate_uom* field. The time unit of measure (e.g. hours or minutes) is in the *run_rate_per* field.

run_rate_per   A Unit of Measure associated with run_rate. E.g. in 19 liters per minute, this value would be "MINUTES".

run_rate_uom   A Unit of Measure associated with run_rate. E.g in 19 liters per minute, this value would be "LITERS".

secondary1   An alternate batch type for this operation.

secondary2   An alternate batch type for this operation.

secondary3   An alternate batch type for this operation.

sequence_dependent_setup_type   The type this routing puts this resource into after performing this operation. For example, a painting machine may put this resource into a state of "WHITE" or "BLACK" which is used in a setup matrix to specify sequence dependent setup time from type "WHITE" to "BLACK" and vice versa. If no type is input, then the type is set to a concatenation of operation and part. This field has meaning only if a setup matrix has been established.

transfer_batch_quantity   Transfer batch quantity for the operation.

preop_time   The amount of time materials occupy the resource before runtime. The amount of time is independent of the quantity of parts. However, if there are no input parts, this amount of time is ignored.

preop_time_uom   The units used for *preop_time* .

unit_runtime   Specifies the runtime per unit of *consumed_part_number* specified in the file containing records of *Bill_Of_Materials_Record* . For instance, if *operation* "Assemble_Table" requires "8" "legs" and "2" "table_top" and *unit_runtime* and *unit_runtime_uom* are "1" "HOURS" to produce 2 tables, and we plan an order for three tables (12 legs and 3 table_tops), the total runtime of the operation is 1.5 hours.

unit_runtime_uom   the unit of measure for *unit_runtime* .

4.50   Sequence_Dependent_Setup_Time_Record

Each record specifies the sequence dependent setup time for changing a resource from processing operations of one type to operations of another type.

A matrix is used to define a set of setup times. A resource relies on a particular setup matrix. Any vertices left undefined in the setup matrix default to a value specified in a record of type *Resource_Record* associated with the particular resource.

This file is readable.

Sequence_Dependent_Setup_Time_Record				
Field	Req	Class	Type	Default
setup_matrix_id	P	Setup_Matrix	Name	-Required-
sequence_dependent_setup_type_from		SD_Setup_Type	Name	UNKNOWN_SETUP_TYPE
sequence_dependent_setup_type_to		SD_Setup_Type	Name	UNKNOWN_SETUP_TYPE
setup_time_uom	1	Unit_Of_Measure	Name	BLANK_UOM
setup_time	1	Scalar_Time	number	0

sequence_dependent_setup_type_from   The name of a source setup type, e.g. "green paint".

sequence_dependent_setup_type_to   The name of a destination setup type, e.g. "purple paint". It is an error when this field is the same as *sequence_dependent_setup_type_from* .

setup_matrix_id   The name of a setup matrix. This name is referenced by different resources via the file containing records of type *Resource_Record* .

setup_time   The time it takes to setup if a routing previously left a resource in a state of *sequence_dependent_setup_type_from* and now needs to change to *sequence_dependent_setup_type_to* .

setup_time_uom   The units used for "setup_time".

4.51 Shift_Record

Records of this file define shift information that is read into into *Rhythm* .

This file is readable.

Shift_Record				
Field	Req	Class	Type	Default
location		String		NULL
shift_id	P	String		-Required-
start_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
start_time		Formatted_Time	Formatted_Time	Unknown_Time
end_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
end_time		Formatted_Time	Formatted_Time	Unknown_Time

end_time   The end time for the shift.

end_time_format   Date/time format to use for end_time.

location   location where this shift is valid. If location is not specified - this shift is assumed to be a globaly defined shift valid for all locations.

shift_id   id to identify the shift.

start_time   The start time for the shift.

start_time_format   Date/time format to use for start_time.

4.52   Short_Late_Orders_Record

This file contains a list of short and late orders in the plan generated by *Rhythm* .

This file is readable.

Short_Late_Orders_Record				
Field	**Req**	**Class**	**Type**	**Default**
demand_order_id		Name	Name	NULL
type		Pruned_String	Pruned_String	NULL
part_number	U	Part_Number	Name	
order_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
order_quantity	U	Part_Quantity	number	
available_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
available_time		Formatted_Time	Formatted_Time	Unknown_Time
amount_late_uom		Unit_Of_Measure	Name	BLANK_UOM
amount_late		Scalar_Time	number	0
reason		Pruned_String	Pruned_String	NULL

amount_late	The amount by which the order is late.
amount_late_uom	The time unit of measure in which the *amount_late* is specified.
available_time	The time the order is completed.
available_time_format	Date/time format to use for available_time.
demand_order_id	The name of the demand order.
part_number	The part being produced by this demand order.
order_quantity	The quantity of parts in units of *output_quantity_uom* ordered for this demand order.
order_quantity_uom	The Unit of Measure associated with order_quantity.
reason	
type	*SHORT* for reporting short orders, and *LATE* for reporting late orders.

4.53 Super_Order_Mapping_Record

This record contains a record of all the demand order splits that have been done in *Rhythm* . This file will enable *Rhythm* to recreate the splits while reading in a saved plan.

This file is readable.

Super_Order_Mapping_Record				
Field	Req	Class	Type	Default
super_order		Name	Name	NULL
sub_order		Name	Name	NULL
sub_order_qty	U	Part_Quantity	number	

sub_order   One of many split orders.

sub_order_qty   Output quantity of the split sub-order.

super_order   When an order is split, a super order is created in *Rhythm* . It keeps track of all the split sub-orders.

4.54 Supplier_Part_Record

Records of this file are used to define fixed lead times for parts obtained from vendors. For a specific vendor, lead times for each part number can be defined. Multiple vendors can be defined for a part, in which case the vendor with the earliest lead time will be selected.

This file is readable.

Supplier_Part_Record				
Field	Req	Class	Type	Default
vendor	P	Vendor	Name	-Required-
part_number	P	Vendor_Part	Name	-Required-
lead_time_uom		Unit_Of_Measure	Name	BLANK_UOM
lead_time		Scalar_Time	number	0
cost		Money	number	0
cost_uom		Unit_Of_Measure	Name	BLANK_UOM
min_quantity_uom		Material_UOM	Name	Material_Quantity_UO
min_quantity		Part_Quantity	number	0
part_quantity_uom		Material_UOM	Name	Material_Quantity_UO
max_quantity		Vendor_Max_Quantity	Vendor_Max_Quantity	Infinity
min_lot_size		Procurement_Min_Lot_Size	Procurement_Min_Lot_Size	0
max_lot_size		Procurement_Max_Lot_Size	Procurement_Max_Lot_Size	Infinity
inc_lot_size		Procurement_Inc_Lot_Size	Procurement_Inc_Lot_Size	1
procurement_time_uom		Unit_Of_Measure	Name	BLANK_UOM
procurement_time		Scalar_Time	number	604800
procurement_horizon_uom		Unit_Of_Measure	Name	BLANK_UOM
procurement_horizon		Scalar_Time	number	SCALAR_TIME_MAX
vendor_part		Name	Name	
count		Integer	Integer	1

cost	Unused
cost_uom	Unused
count	Number of times "max_quantity" will be purchased.
inc_lot_size	if the number of units required for a part falls between minimum and maximum lot size after an initial order has been placed then procurement has to be done in increments of inc_lot_size.
lead_time	The normal or expected lead time for this part.
lead_time_uom	A Unit Of Measure specifies a conversion from some external unit to the internal units used by Rhythm. For example, Rhythm stores all times as seconds. The *days* unit of measure converts days to seconds, while the *hours* unit of measure converts hours to seconds. The default (blank) unit of measure multiplies all values by one.
part_quantity_uom	The material quantity unit of measure in which *material_quantity_uom* is specified.
part_number	The part this vendor can supply. If the value is set to ALL_RAW_MATERIALS, then this record applies to all raw parts in the system. Raw parts are those parts which have no means of being produced (no BOM produces them). This value is used for doing

demos when lead times are not available.

max_lot_size	The maximum quantity of parts that this vendor will supply for a part at a time. if more units of this part are required - the vendor will have to order in multiples of max lot size and min lot size.
max_quantity	The maximum number of parts a vendor can supply within the defined *lead_time* . If left blank. there is no limit on the number of parts a vendor can supply within the given lead time.
min_lot_size	The minimum lot size that this vendor will supply for procurement of a part. The user has to order at least this many units of this part at least for first time time procurement. If the remaining quantity falls within min lot size and max lot size the user will be supplied parts in increments of inc_lot_size.
min_quantity	Unused
min_quantity_uom	The Unit of Measure associated with min_quantity.
procurement_horizon	the horizon beyond which procurement lot sizing is not done.
procurement_horizon_uom	A Unit Of Measure specifies a conversion from some external unit to the internal units used by Rhythm. For example, Rhythm stores all times as seconds. The *days* unit of measure converts days to seconds, while the *hours* unit of measure converts hours to seconds. The default (blank) unit of measure multiplies all values by one.
procurement_time	The horizon or time interval over which procurement lot sizing is done.
procurement_time_uom	A Unit Of Measure specifies a conversion from some external unit to the internal units used by Rhythm. For example, Rhythm stores all times as seconds. The *days* unit of measure converts days to seconds, while the *hours* unit of measure converts hours to seconds. The default (blank) unit of measure multiplies all values by one.
vendor	The name of the vendor for this *part_number* . When *Rhythm* Interplant functionality is used via *Supplier_Record* , *vendor* can be an Interplant plant name supplying *part_number* to this plant.
vendor_part	*Vendor* 's name for *part_number* . This field defaults to *part_number* when blank, meaning *vendor* and this *Rhythm* server both call the part by the same name. This feature is currently only relevant when *Rhythm* Interplant functionality is used.

4.55 Supplier_Record

This file identifies the Interplant suppliers among the part vendors specified in *Supplier_Part_Record* s. An Interplant supplier is a *Rhythm* server in a network of servers connected by Interplant data files which feed each other demands and respond to each other with part reservations. This file also contains one record which specifies the name of *this Rhythm* server as an Interplant consumer (and possibly supplier). It is an error to leave out this record if any Interplant suppliers are specified.

This file is readable.

Supplier_Record				
Field	Req	Class	Type	Default
supplier	P	Name	Name	-Required-
type	P	Name	Name	-Required-
data_directory		Pruned_String	Pruned_String	

data_directory When the *type* field is *INTERPLANT* or *SELF* this field is a string providing the path to the data directory for *supplier* 's rhythm_server. When *type* is *SELF* , *data_directory* defaults to *DIR/* where DIR is determined by the -dir command line option. When *type* is *INTERPLANT* , *data_directory* defaults to *DIR/../SUPPLIER/* where DIR is the data_directory for the *SELF* record and SUPPLIER is the *supplier* field of the file. Note that the pathname supplied should have a trailing / .

supplier The meaning of this field is determined by the *type* field. If *type* is *SELF* , *supplier* is the name of this *Rhythm* server (plant) in the Interplant network. If *type* is *INTERPLANT* , *supplier* is the name of another *Rhythm* server (plant) in the Interplant network which can supply this server parts specified in supplier_part_data. Part demands and responses are communicated through Interplant data files. If *type* is *VENDOR* , *supplier* is an external (non-Interplant) vendor supplying parts specified in supplier_part_data (currently such records are ignored).

type One of the strings *INTERPLANT* , *SELF* or *VENDOR* . Each value specifies a different interpretation for the *vendor* field. If any *INTERPLANT* records occur in the file, one record must be of type *SELF* .

4.56 Transportation_Time_Record

Each record specifies the time it takes to transport parts from *from_location* to *to_location* . Transportation times are a critical factor determining the time between each planned operation.

This file is readable.

Transportation_Time_Record				
Field	Req	Class	Type	Default
part_number		Name	Name	NULL
from_location	P	Location	Name	-Required-
to_location	P	Location	Name	-Required-
transportation_time_uom	P	Unit_Of_Measure	Name	-Required-
transportation_time	P	Scalar_Time	number	-Required-

from_location A location, commonly defined in the file containing records of *Resource_Record* . This is the source location.

part_number The name of a part for this particular transportation time. If this part is left blank, the default transportation time between *from_location* and *to_location* is defined. If certain parts require additional transportation between the same two locations, another record with a part number is defined, overriding the default transportation time. This field is currently not used.

to_location A location, commonly defined in file containing records of *Resource_Record* . This is the destination location.

transportation_time The time transportation of material takes between *from_location* and *to_location* . This only defines the time in the forward direction. For example, if *from_location* is A and *to_location* is B, the time from A — B is defined, but the time from B — A is not defined (and is therefore zero).

transportation_time_uom The time units used for *transportation_time* .

4.57 Unassigned_Inventory_Record

Each record specifies a quantity of a part not yet assigned to any demand orders or manufacturing orders. These parts are available for *Rhythm* to assign to demand orders and manufacturing orders. They contrast with the parts specified in wip_data which are parts already assigned to orders, and parts in vendor_data which can be procured from vendors.

This file is readable.

Unassigned_Inventory_Record				
Field	**Req**	**Class**	**Type**	**Default**
inventory_buffer		Inventory_Buffer	Name	unspecified_inventory_buffer_id
part_number	P	Part_Number	Name	-Required-
part_quantity_uom		Material_UOM	Name	Material_Quantity_UOM
part_quantity		Vendor_Max_Quantity	Vendor_Max_Quantity	Infinity
purchase_order_number	1	Name	Name	NULL
vendor	-1	Vendor	Name	NULL
arrival_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
arrival_time		Formatted_Time	Formatted_Time	Unknown_Time

arrival_time	The time at which this part will be available in *inventory_buffer* . If blank, which is normally the case for at least one record of the file, this field defaults to the current time.
arrival_time_format	The date/time format to use in parsing *arrival_time* .
inventory_buffer	The holding queue in which this part is sitting. If left blank it defaults to the buffer specified by command line option -unspecified_inventory_buffer_id.
part_quantity	The total quantity of *part_number* that is either already present, or will arrive at *arrival_time* . If left blank, the quantity is infinite.
part_quantity_uom	The unit of measure in which *part_quantity* is specified.
part_number	The part number for the quantity of unassigned parts that are being specified.
purchase_order_number	An identifier for the purchase order associated with this part arrival.
vendor	The name of the vendor (provider) for this unassigned part.

4.58 Unassigned_Wip_Record

This file is used to specify parts in process on the shop floor which are not assigned to manufacturing orders. *Rhythm* will make such assignments. The format is similar to Wip_Record. However, instead of referencing a manufacturing order, several of the fields identify the particular bill of material, routing, and operation processing the parts.

This file is readable.

Unassigned_Wip_Record				
Field	Req	Class	Type	Default
routing	P	Routing	Name	-Required-
produced_part	P	Part_Number	Name	-Required-
operation	P	Name	Name	-Required-
ecn_date_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
ecn_date		Formatted_Time	Formatted_Time	Unknown_Time
unfinished_part	1	Part_Number	Name	NULL
unfinished_part_quantity_uom	-1	Material_UOM	Name	Material_Quantity_UOM
unfinished_part_quantity	-1	Part_Quantity	number	0
finished_part	2	Part_Number	Name	NULL
finished_part_quantity_uom	-2	Material_UOM	Name	Material_Quantity_UOM
finished_part_quantity	-2	Part_Quantity	number	0

ecn_date   The effective date of the bill of material producing the unassigned wip. Usually the effective date is *current time*, represented by a blank.

ecn_date_format   Date/time format to use for ecn_date.

finished_part   The output part of this operation, usually identical to produced_part.

finished_part_quantity   The unassigned wip quantity of *finished_part* currently located after this operation. This quantity has not yet moved to downstream operations.

finished_part_quantity_uom   The Unit of Measure associated with finished_part_quantity

operation   The operation within the routing where this unassigned wip is located.

produced_part   The bill of material output part, usually identical to finished_part.

routing   The routing currently processing this unassigned wip.

unfinished_part   An input part of this operation.

unfinished_part_quantity   The unassigned wip quantity of *unfinished_part* currently located in front of this operation.

unfinished_part_quantity_uom   The Unit of Measure associated with unfinished_part_quantity.

4.59 Unit_Of_Measure_Record

Defines new units of measure and conversion factors between units of measure. Built-in types include "Material_Quantity", "Money", "Space", and "Scalar_Time". All conversions must ultimately be anchored in a built-in type. Additional built-in types include "WEEKS", "DAYS", "HOURS", "MINUTES" and "SECONDS".

Units of Measure are reflexive and transitive. This means that defining a conversion from minutes to seconds automatically defines the conversion from seconds to minutes. It also means that defining conversions from minutes to seconds and hours to minutes automatically defines a conversion from hours to seconds.

This file is readable.

Unit_Of_Measure_Record				
Field	Req	Class	Type	Default
part		Part_Number	Name	NULL
from_uom	P	Name	Name	-Required-
to_uom	P	Name	Name	-Required-
conversion	P	number	number	-Required-

conversion   Conversion factor from *from_uom* to *to_uom* . E.g. "24".

from_uom   Name of a unit to convert from. E.g. "DAYS".

part   Part number associated with the unit of measure

to_uom   Name of a unit to convert to. E.g. "HOURS".

4.60   Use_Effectivity_Mutations_Record

This file is maintained by *Rhythm* for use in saving and restoring plans. *Rhythm* replaces any Use-Up and Accompanying Parts having the *ecn_number* for this record with any Replacement Parts having the *ecn_number* for this record. For instance, a given BOM might specify an *ecn_number* of 100 with Use-Up Parts 10 and 11. Accompanying Part 12, and Replacement Parts 13, 14, and 15. It could also specify an *ecn_number* of 101 with Use-Up Part 20, no Accompanying Parts, and Replacement Part 21. If *Rhythm* runs out of part 10, it replaces demand for 10, 11, and 12 with parts 13, 14, and 15. Later, if it runs out of part 20, it starts using 21 instead.

This file is output by Rhythm. The *unmutated_portion* is the percentage of parts used as Use-Up Parts. The remaining percentage is those parts used as Replacement Parts.

This file is readable.

Use_Effectivity_Mutations_Record				
Field	Req	Class	Type	Default
mfg_order	P	Manufacturing_Order	Name	-Required-
ecn_number	P	String		-Required-
unmutated_portion	P	number	number	-Required-

ecn_number	The tag which binds sets of Use-Up, Accompanying, and Replacement Parts (see *ecn_code*). For a given *ecn_number* , its Use-Up and Accompanying Parts are replaced by the Replacement Parts. See *Bill_Of_Materials_Record* .
mfg_order	A unique id used to identify a particular manufacturing order.
unmutated_portion	Percentage of parts used as Use-Up Parts.

4.61 Variable_Capacity_Bucket_Size_Record

This record allows the definition of variable sized time buckets used for rough scheduling purposes and many graphical display elements (e.g. load graphs). The overall scheduling horizon is affected by the size of buckets defined and the number of each bucket size defined.

This file is readable.

Variable_Capacity_Bucket_Size_Record				
Field	Req	Class	Type	Default
bucket_size_uom	P	Unit_Of_Measure	Name	-Required-
bucket_size	P	Scalar_Time	number	-Required-
number_of_buckets_of_given_size		Integer	Integer	1
rollover_period_uom	1	Unit_Of_Measure	Name	BLANK_UOM
rollover_period	1	Scalar_Time	number	0

bucket_size   The number of buckets of size *bucket_size_uom* .

bucket_size_uom   The units used for *bucket_size* .

number_of_buckets_of_given_size   The total number of buckets of the defined size.

rollover_period   The period of time between refreshes of the capacity bucket sizes. For instance, if have 7 DAY buckets and 12 WEEK buckets and rollover period is one week, then as the planner advances time, it chops off the 7 day buckets until one week (7 day buckets) is gone. Then it will generate a fresh set of 7 day buckets and 12 week buckets.

rollover_period_uom   A Unit Of Measure specifies a conversion from some external unit to the internal units used by Rhythm. For example, Rhythm stores all times as seconds. The *days* unit of measure converts days to seconds, while the *hours* unit of measure converts hours to seconds. The default (blank) unit of measure multiplies all values by one.

4.62   Wip_Assignments_Output_Record

This file is used to output *Rhythm* 's assignments of unassigned_wip_data to manufacturing orders. To make the assignments persist between runs, use the report to convert records of unassigned_wip_data into wip_data. The file does not include assignments specified to *Rhythm* through file wip_data.

This file is readable.

Wip_Assignments_Output_Record				
Field	Req	Class	Type	Default
manufacturing_order	U	Manufacturing_Order	Name	
operation		Name	Name	NULL
unfinished_part		Part_Number	Name	NULL
unfinished_part_uom		Material_UOM	Name	Material_Quantity_UOM
unfinished_part_quantity		Part_Quantity	number	0
finished_part		Part_Number	Name	NULL
finished_part_uom		Material_UOM	Name	Material_Quantity_UOM
finished_part_quantity		Part_Quantity	number	0

finished_part   An output part of the operation.

finished_part_quantity   The quantity of *finished_part* located after the operation and assigned to *manufacturing_order* .

finished_part_uom   The Unit of Measure associated with finished_part_quantity.

manufacturing_order   The manufacturing order which received the wip assignments.

operation   The operation within the routing where the wip is assigned.

unfinished_part   An input part of the operation.

unfinished_part_quantity   The quantity of *unfinished_part* in front of the operation and assigned to *manufacturing_order* .

unfinished_part_uom   The Unit of Measure associated with unfinished_part_quantity.

4.63 Wip_Record

This file is used to specify parts on the shop floor which are assigned to a given manufacturing order. It is also used to specify when an operation of a given manufacturing order is complete.

For this release of the software, *unfinished_inventory_buffer* and *finished_inventory_buffer* should be blank, specifying a global *default buffer* .

unfinished_part_number and *unfinished_quantity* specify parts not yet processed by *operation* . *finished_part_number* and *finished_quantity* specify parts already processed by *operation* but not yet transported to the next operation. If both are zero, there is no reason to supply the record unless you want to mark *operation_complete_p* as being true.

Since a given operation can have several input parts, several records for a given pair of manufacturing_order and operation can occur.

Some users will have trouble reporting *unfinished_quantity* but can report them in terms of finished parts. For instance, if the operation input part quantities are "1" "PartA" and "2" "PartB" and the output part quantities are "1" "PartC", the user might not be able to report that a manufacturing order has unfinished "10" "PartA" and unfinished "20" "PartB" at the first operation. Instead, the user can report these in terms of output part "PartC" by saying that the first operation has "10" unfinished "PartC"'s.

This file is readable.

Wip_Record				
Field	Req	Class	Type	Default
manufacturing_order	P	Name	Name	-Required-
operation	P	Name	Name	-Required-
operation_sequence_number		Integer	Integer	-1
specific_routing_id		Part_Number	Name	NULL
unfinished_part_number	1	Part_Number	Name	NULL
unfinished_quantity_uom	-1	Material_UOM	Name	Material_Quantity_UOM
unfinished_quantity	-1	Part_Quantity	number	0
unfinished_inventory_buffer	-1	Inventory_Buffer	Name	unspecified_inventory_buffer_id
finished_part_number	2	Part_Number	Name	NULL
finished_quantity_uom	-2	Material_UOM	Name	Material_Quantity_UOM
finished_quantity	-2	Part_Quantity	number	0
finished_inventory_buffer	-2	Inventory_Buffer	Name	unspecified_inventory_buffer_id
start_time_format	-3	Time_Format	Time_Format	DDMMMYYYYhhmmss
start_time	3	Formatted_Time	Formatted_Time	Unknown_Time
elapsed_runtime_uom	-4	Unit_Of_Measure	Name	BLANK_UOM
elapsed_runtime	4	Scalar_Time	number	unknown.FLOAT
completion_time_format		Time_Format	Time_Format	DDMMMYYYYhhmmss
completion_time		Formatted_Time	Formatted_Time	Unknown_Time
operation_complete_p		Boolean	char	FALSE
processed_quantity	-2	Part_Quantity	number	0
acm_flag		ACM_Flag	char	ACM_Add

acm_flag Add Cancel Modify flag is a single character field containing one of the characters "A" "C" "M" or " ". *Blank* and nothing at all default to *Add* .

completion_time The time this WIP was complete.

completion_time_format Date/time format to use for completion_time.

elapsed_runtime	The elapsed runtime that should be consumed from the operation runtime. Users can specify either *elapsed_runtime* or *start_time*. If start_time is specified *elapsed_runtime* is calculated to be *elapsed_runtime = planner current_time - start_time*.
elapsed_runtime_uom	The time unit of measure in which the *elapsed_runtime* is specified.
finished_inventory_buffer	The location where *finished_part_number* is located. This field is currently unused.
finished_quantity	The quantity of *finished_part_number* produced and located after this operation on the shop floor at the time this record is read by *Rhythm*.
finished_quantity_uom	The Unit of Measure associated with finished_material_quantity
finished_part_number	The part number produced at this operation by this manufacturing order.
manufacturing_order	The unique manufacturing order id used to associate this WIP with a particular demand order.
operation	The operation where this WIP is at within a particular routing.
operation_complete_p	A flag defining whether this WIP is complete or not. A "T" here means the specified operation of the manufacturing order has been processed. An "F" means the specified operation of the manufacturing order is not finished yet.
operation_sequence_number	This field is obsolete.
processed_quantity	The total quantity of *finished_part_number* this operation has ever completed. This does not describe inventory currently positioned after the operation. It is a historical total. This data is used mostly for display purposes. However, at the last operation of a final assembly manufacturing order, the quantity is treated as additional shippable parts. This helps in two wip reporting scenarios: 1. customers who report only unfinished_quantity and processed_quantity. If they do not have a way of reporting finished_quantity at the last operation, the processed_quantity at the last operation will cover reporting of the finished goods. 2. customers who ship out the finished goods in batches (e.g. truckloads). They can utilize both finished_quantity and processed_quantity. Before any goods are shipped, finished_quantity holds the total quantity. As they ship portions, they can move quantities from finished_quantity to processed_quantity. Without doing this, *Rhythm* would treat such orders are short and would try to fix the shortages.
specific_routing_id	obsolete field
start_time	The time this WIP was started. Users can either specify *start_time* or the *elapsed_runtime*. If *elapsed_time* is specified the *start_time* is assumed to be *start_time = planner current_time - elapsed_time*.
start_time_format	Date/time format to use for start_time.
unfinished_inventory_buffer	The location of the *unfinished_part_number*. This field is currently unused.
unfinished_quantity	The quantity of *unfinished_material_type* yet to be processed.

unfinished_quantity_uom   The Unit of Measure associated with unfinished_material_quantity

unfinished_part_number   A part waiting to be processed at this operation.

Index